

§ 14. 运算符重载 - 转换构造和类型转换函数的使用



要求:

- 1、完成本文档中所有的题目并写出分析、运行结果
- 2、无特殊说明，均使用 **VS2019 + Linux** 编译
- 3、直接在本文件上作答，**写出答案/截图（不允许手写、不允许手写拍照）**即可；填写答案时，为适应所填内容或贴图，**允许调整**页面的字体大小、颜色、文本框的位置等
★ 在保证一页一题的前提下，具体页面布局可以自行发挥，简单易读即可
- 4、转换为pdf后提交
- 5、**4月22日前**网上提交本次作业（在“实验报告”中提交）



§ 14. 运算符重载 - 转换构造和类型转换函数的使用

填写要求:

1、依次用VS2019/Linux编译指定的源程序文件

如果编译正确，则对应位置填写运行结果并给出得到此结果的原因解释

如果编译错误，则对应位置填写该行的编译错误提示及错误原因分析

2、如果编译器报多个错误，填写源程序文件对应行的错误提示即可，示例如下，将红色框截图即可

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      cout << "Hello, world!\n" << endl;
7      return 0;
8  }
9
```

输出

显示输出来源(S): 生成

1>----- 已启动生成: 项目: demo, 配置: Debug Win32 -----

1> demo.cpp

1>d:\demo\demo\demo.cpp(7): error C2143: 语法错误: 缺少“;” (在“return”的前面)

===== 生成: 成功 0 个, 失败 1 个, 最新 0 个, 跳过 0 个 =====

test.cpp: 在函数 int main() 中:

test.cpp:7:2: 错误: expected ';' before 'return'

return 0;

```
c3 = c1 + Complex(2.5); //假设此句错误
c3.display();
```

```
// c3 = c1 + Complex(2.5);
// c3.display();
```

3、如果main函数中某一句错误，则将该句及下面的打印语句全部注释掉，继续观察其余正确语句的运行结果（示例见上图）

4、用蓝色加粗字体填写

5、不需要填写的部分可以删除（例如：某句正确，则错误部分不填，或填写/即可）



§ 14. 运算符重载 - 转换构造和类型转换函数的使用

附：用WPS等其他第三方软件打开PPT，将代码复制到VS2019中后，如果出现类似下面的**编译报错**，则观察源程序编辑窗

的右下角是否为CR，如果是，单击CR，在弹出中选择CRLF，再次CTRL+F5运行即可





§ 14. 运算符重载 - 转换构造和类型转换函数的使用

1. 无转换构造函数、无类型转换函数、友元方式实现复数+

c3 = c1 + Complex(2.5)

编译正确，运行结果：_____，能得到此结果的原因：_____

编译错误，VS2019下：error C2440: “<function-style-cast>”: 无法从“double”转换为“Complex” (删除横线后贴图)

Linux下：错误：no matching function for call to ‘Complex::Complex(double)’ (删除横线后贴图)

错误原因分析：___无以一个double为参的Complex构造函数或能对应的默认参数_____

c3 = c1 + 2.5

编译正确，运行结果：_____，能得到此结果的原因：_____

编译错误，VS2019下：error C2679: 二元“+”: 没有找到接受“double”类型的右操作数的运算符(或没有可接受的转换) (后贴图)

Linux下：错误：no match for ‘operator+’ (operand types are ‘Complex’ and ‘double’)

错误原因分析：___无Complex+double的重载；无Complex转double、无法double+；无double转Complex、无法Complex+_____

c3 = 2.5 + c1

编译正确，运行结果：_____，能得到此结果的原因：_____

编译错误，VS2019下：error C2678: 二进制“+”: 没有找到接受“double”类型的左操作数的运算符(或没有可接受的转换) (贴图)

Linux下：错误：no match for ‘operator+’ (operand types are ‘Complex’ and ‘double’)

错误原因分析：无double+Complex重载；无Complex转double，无法double+；无double转Complex、无法Complex+_____

编译错!!!

- 1、无 double+复数 的重载
- 2、无 复数转double 的类型转换函数，也无法理解为 double+
- 3、无 double转复数 的转换构造函数及 复数+复数 的重载，也无法理解为 复数+

2.5 + c1，
因为没有定义double+复数的重载，
因此c1被转换为double
(隐式调用类型转换函数)，
再double相加，得5.5

★ 每个输出的不匹配项可删除（例：若本项编译正确，则编译错误的几行内容直接删除即可）

★ 分析正确/错误原因时，仿课件P. 45~46的样式，需要将正确/错误原因交待清楚

★ 后续页面要求相同



§ 14. 运算符重载 - 转换构造和类型转换函数的使用

2. 无转换构造函数、无类型转换函数、成员方式实现复数+

`c3 = c1 + Complex(2.5)`

编译正确，运行结果：_____，能得到此结果的原因：_____

编译错误，VS2019下：error C2440: “<function-style-cast>”: 无法从“double”转换为“Complex” (可删除横线后贴图)

Linux下：错误：no matching function for call to ‘Complex::Complex(double)’ (贴图)

错误原因分析：___无以一个double为参的Complex构造函数或能对应的默认参数_____

`c3 = c1 + 2.5`

编译正确，运行结果：_____，能得到此结果的原因：_____

编译错误，VS2019下：error C2679: 二元“+”: 没有找到接受“double”类型的右操作数的运算符(或没有可接受的转换) (贴图)

Linux下：错误：no match for ‘operator+’ (operand types are ‘Complex’ and ‘double’)

错误原因分析：___无Complex+double的重载；无Complex转double、无法理解为double+；无double构Complex，无法理解为Complex+_____

`c3 = 2.5 + c1`

编译正确，运行结果：_____，能得到此结果的原因：_____

编译错误，VS2019下：error C2677: 二进制“+”: 没有找到接受“Complex”类型的全局运算符(或没有可接受的转换) (贴图)

Linux下：错误：no match for ‘operator+’ (operand types are ‘Complex’ and ‘double’)

错误原因分析：___无double+Complex的重载；无Complex转double、无法理解为double+；无其它自定义相关全局重载_____



§ 14. 运算符重载 - 转换构造和类型转换函数的使用

3. 无转换构造函数、有类型转换函数、友元方式实现复数+

`c3 = c1 + Complex(2.5)`

编译正确，运行结果：_____，能得到此结果的原因：_____

编译错误，VS2019下：error C2440: “<function-style-cast>”: 无法从“double”转换为“Complex” (可删除横线后贴图)

Linux下：错误: no matching function for call to ‘Complex::Complex(double)’ (删除横线后贴图)

错误原因分析：_____无以一个double为参的Complex构造函数或能对应的默认参数_____

`c3 = c1 + 2.5`

编译正确，运行结果：_____，能得到此结果的原因：_____

编译错误，VS2019下：error C2679: 二元“=”: 没有找到接受“double”类型的右操作数的运算符(或没有可接受的转换) (贴图)

Linux下：错误: no match for ‘operator=’ (operand types are ‘Complex’ and ‘double’)

错误原因分析：无Complex+double; 有Complex转double, c1+2.5为double->无Complex=double; ; 无double构Complex, 无法Complex+Complex

`c3 = 2.5 + c1`

编译正确，运行结果：_____，能得到此结果的原因：_____

编译错误，VS2019下：error C2679: 二元“=”: 没有找到接受“double”类型的右操作数的运算符(或没有可接受的转换) (贴图)

Linux下：错误: no match for ‘operator=’ (operand types are ‘Complex’ and ‘double’)

错误原因分析：无double+Complex; 有Complex转double, 2.5+c1为double->无Complex=double; ; 无double构Complex, 无法Complex+Complex



§ 14. 运算符重载 - 转换构造和类型转换函数的使用

4. 无转换构造函数、有类型转换函数、成员方式实现复数+

`c3 = c1 + Complex(2.5)`

编译正确，运行结果：_____，能得到此结果的原因：_____

编译错误，VS2019下：error C2440: “<function-style-cast>”: 无法从“double”转换为“Complex” (可删除横线后贴图)

Linux下：错误: no matching function for call to ‘Complex::Complex(double)’ (贴图)

错误原因分析：___无以一个double为参的Complex构造函数或能对应的默认参数___

`c3 = c1 + 2.5`

编译正确，运行结果：_____，能得到此结果的原因：_____

编译错误，VS2019下：error C2679: 二元“=”: 没有找到接受“double”类型的右操作数的运算符(或没有可接受的转换) (贴图)

Linux下：错误: no match for ‘operator=’ (operand types are ‘Complex’ and ‘double’) (图)

错误原因分析：_无Complex+double; 有Complex转double, c1+2.5为double -> 无Complex=double; ; 无double转Complex, 无法Complex+Complex

`c3 = 2.5 + c1`

编译正确，运行结果：error C2679: 二元“=”: 没有找到接受“double”类型的右操作数的运算符(或没有可接受的转换)

编译错误，VS2019下：错误: no match for ‘operator=’ (operand types are ‘Complex’ and ‘double’)

Linux下：_____ (可删除横线后贴图)

错误原因分析：_无double+Complex; 有Complex转double, 2.5+c1为double->无Complex=double; ;



§ 14. 运算符重载 - 转换构造和类型转换函数的使用

5. 有转换构造函数、无类型转换函数、友元方式实现复数+

`c3 = c1 + Complex(2.5)`

编译正确, 运行结果: `5.5+4i 5.5+4i`, 能得到此结果的原因:

有`double`构造`Complex` -> 有唯一`Complex+Complex`, 为`Complex` -> 有默认`Complex=Complex`, 运算出来就是 $(3, 4) + (2.5, 0) = (5.5, 4)$ 。

编译错误, VS2019下: _____ (可删除横线后贴图)

Linux下: _____ (可删除横线后贴图)

错误原因分析: _____

`c3 = c1 + 2.5`

编译正确, 运行结果: `5.5+4i 5.5+4i`, 能得到此结果的原因:

无`Complex+double`; 无`Complex`转`double`; 有`double`构造`Complex`, 且唯一重载, 变成 `c3=c1+Complex(2.5)`, 之后同上

编译错误, VS2019下: _____ (可删除横线后贴图)

Linux下: _____ (可删除横线后贴图)

错误原因分析: _____

`c3 = 2.5 + c1`

编译正确, 运行结果: `5.5+4i 5.5+4i`, 能得到此结果的原因:

无`double+Complex`; 无`Complex`转`double`; 加号有唯一匹配 `Complex+Complex`, 且有唯一`double`构造`Complex`, 变成 `c3=Complex(2.5)+c1`

编译错误, VS2019下: _____ (可删除横线后贴图)

Linux下: _____ (可删除横线后贴图)

错误原因分析: _____



§ 14. 运算符重载 - 转换构造和类型转换函数的使用

6. 有转换构造函数、无类型转换函数、成员方式实现复数+

`c3 = c1 + Complex(2.5)`

编译正确，运行结果：`5.5+4i 5.5+4i`，能得到此结果的原因：

有`double`构`Complex` -> 有唯一`Complex+Complex`，为`Complex` -> 有默认`Complex=Complex`，运算出来就是 $(3, 4) + (2.5, 0) = (5.5, 4)$ 。

编译错误，VS2019下：_____（可删除横线后贴图）

Linux下：_____（可删除横线后贴图）

错误原因分析：_____

`c3 = c1 + 2.5`

编译正确，运行结果：`5.5+4i 5.5+4i`，能得到此结果的原因：

无`Complex+double`；无`Complex`转`double`；有`double`构`Complex`，且唯一重载，变成 `c3=c1+Complex(2.5)`，之后同上

编译错误，VS2019下：_____（可删除横线后贴图）

Linux下：_____（可删除横线后贴图）

错误原因分析：_____

`c3 = 2.5 + c1`

编译正确，运行结果：_____，能得到此结果的原因：_____

编译错误，VS2019下：`error C2677: 二进制“+”：没有找到接受“Complex”类型的全局运算符(或没有可接受的转换)`后贴图

Linux下：__错误：no match for ‘operator+’ (operand types are ‘double’ and ‘Complex’)

错误原因分析：__无`double+Complex`；无`Complex`转`double`，无法`double+`；无全局自定义能对应的加法重载(成员函数运算符重载，左值（自己）不能是隐式类型转换得来的？)



§ 14. 运算符重载 - 转换构造和类型转换函数的使用

7. 有转换构造函数、有类型转换函数、友元方式实现复数+

`c3 = c1 + Complex(2.5)`

编译正确，运行结果：`5.5+4i 5.5+4i`，能得到此结果的原因：

有 `Complex(double)`，有唯一 `Complex+Complex`，为 `Complex`，有默认 `Complex=Complex`。

编译错误，VS2019下：_____（可删除横线后贴图）

Linux下：_____（可删除横线后贴图）

错误原因分析：_____

`c3 = c1 + 2.5`

编译正确，运行结果：_____，能得到此结果的原因：_____

编译错误，VS2019下：`error C2666: "operator +": 2 个重载有相似的转换`（可删除横线后贴图）

Linux下：错误：ambiguous overload for 'operator+' (operand types are 'Complex' and 'double')

错误原因分析：__无 `Complex+double`；；有 `_Complex` 转 `double` 满足 `double+double`，也有 `double` 构 `Complex` 满足 `Complex+Complex`，重载不唯一

`c3 = 2.5 + c1`

编译正确，运行结果：_____，能得到此结果的原因：_____

编译错误，VS2019下：`error C2666: "+" : 2 个重载有相似的转换`（可删除横线后贴图）

Linux下：错误：ambiguous overload for 'operator+' (operand types are 'double' and 'Complex')

错误原因分析：__无 `double+Complex`；；有 `double` 构 `Complex` 满足 `Complex+Complex`，也有 `Complex` 转 `double` 满足 `double+double`，重载不唯一



§ 14. 运算符重载 - 转换构造和类型转换函数的使用

8. 有转换构造函数、有类型转换函数、成员方式实现复数+

`c3 = c1 + Complex(2.5)`

编译正确, 运行结果: `5.5+4i` | `5.5+4i`, 能得到此结果的原因:

有`Complex(double)`, 有唯一 `Complex+Complex`, 得`Complex`, 然后使用默认 `Complex=Complex`。即 $(3, 4) + (2.5, 0)$

编译错误, VS2019下: _____ (可删除横线后贴图)

Linux下: _____ (可删除横线后贴图)

错误原因分析: _____

`c3 = c1 + 2.5`

编译正确, 运行结果: _____, 能得到此结果的原因: _____

编译错误, VS2019下: `error C2666: "Complex::operator +": 2 个重载有相似的转换` (可删除横线后贴图)

Linux下: `错误: ambiguous overload for 'operator+' (operand types are 'Complex' and 'double')`

错误原因分析: `无Complex+double; ; 有Complex转double满足double+double, 也有double构Complex满足c1的Complex+Complex, 重载不唯`

`c3 = 2.5 + c1`

`5.5+0i` | `5.5+0i`

编译正确, 运行结果: _____, 能得到此结果的原因:

`无double+Complex; ; 有Complex转double满足double+double, 且无其它全局+重载 (2.5不是Complex对象成员函数Complex+Complex)`, 再隐式调用 `Complex(double)`, 再`Complex=Complex`。

即 `double(c1)=3, 2.5+3=5.5, Complex(5.5)=(5.5, 0)`。

编译错误, VS2019下: _____ (可删除横线后贴图)

Linux下: _____ (可删除横线后贴图)

错误原因分析: _____



§ 14. 运算符重载 - 转换构造和类型转换函数的使用

9. 有转换构造函数、有类型转换函数、成员方式实现复数+、另有double+Complex

5.5+4i

仅讨论语句 `c3 = 2.5 + c1`，回答下列问题

1、为什么编译不错？

因为有唯一的参数个数、类型完全一致的+定义，即 `double+Complex`。

2、运行结果是多少？

5.5+4i

```
//double+复数的重载，只能是友元函数方式
Complex operator+(const double &d1, const Complex &c2)
{
    return Complex(d1+c2.real, c2.imag);
}
```

3、为什么和8的结果不同？

因为这里用的是这里才有的 `double+Complex` 重载，这个函数算 `2.5+(3, 4)` 得到的就是 `(5.5, 4)`。

8那里用的是c1先转为double，丢掉了虚部，再double相加，再转换构造成Complex。



§ 14. 运算符重载 - 转换构造和类型转换函数的使用

10. 单独讨论有类型转换的情况下，cout重载的输出结果与期望值不同

目前main函数中第4个输出语句与期望值不同，原因是：

重载的<<接受的是Complex引用，而加法返回的是一个临时变量，不给引用。

所以这个 cout<<(a+b) 匹配成了 cout << (double(a)+double(b))

仅允许改动两行，使程序输出与期望值相同：

改动第__33__行，原内容： `21 Complex operator+(const Complex &a) //Complex加`

新内容： `21 Complex& operator+(const Complex &a) //Complex加`

改动第__36__行，原内容： `23 Complex c;`

新内容： `23 static Complex c;`

或者？：

改动第__21__行，原内容： `33 friend ostream & operator<<(ostream &out, Complex &c); //重载<<的声明`

新内容： `33 friend ostream & operator<<(ostream &out, Complex c); //重载<<的声明`

改动第__23__行，原内容： `36 ostream &operator<<(ostream &out, Complex &c)`

新内容： `36 ostream &operator<<(ostream &out, Complex c)`