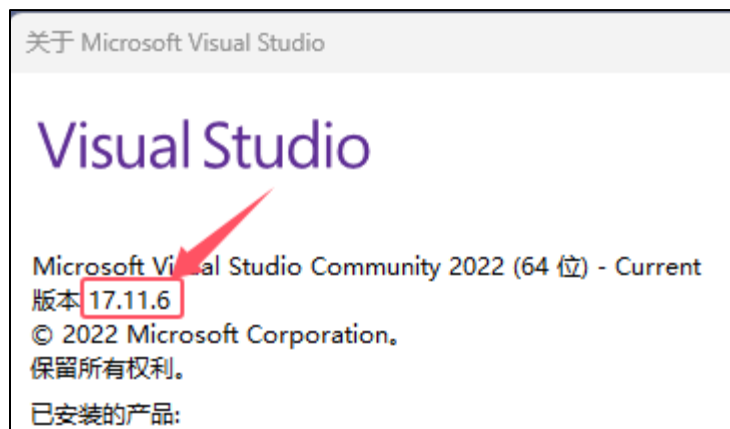


§ 16. 运算符重载 - 转换构造和类型转换函数的使用



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 **VS2022 + 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、**11月28日前**网上提交本次作业（在“文档作业”中提交）
- 6、为了保证一致性，先把VS升级到最新版本后再做（截止2024-11-22的最新版本是17.11.6）





§ 16. 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VS2022/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

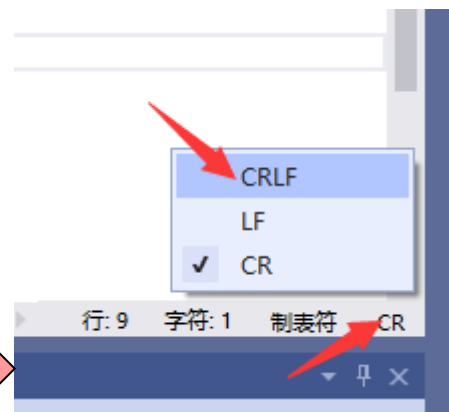
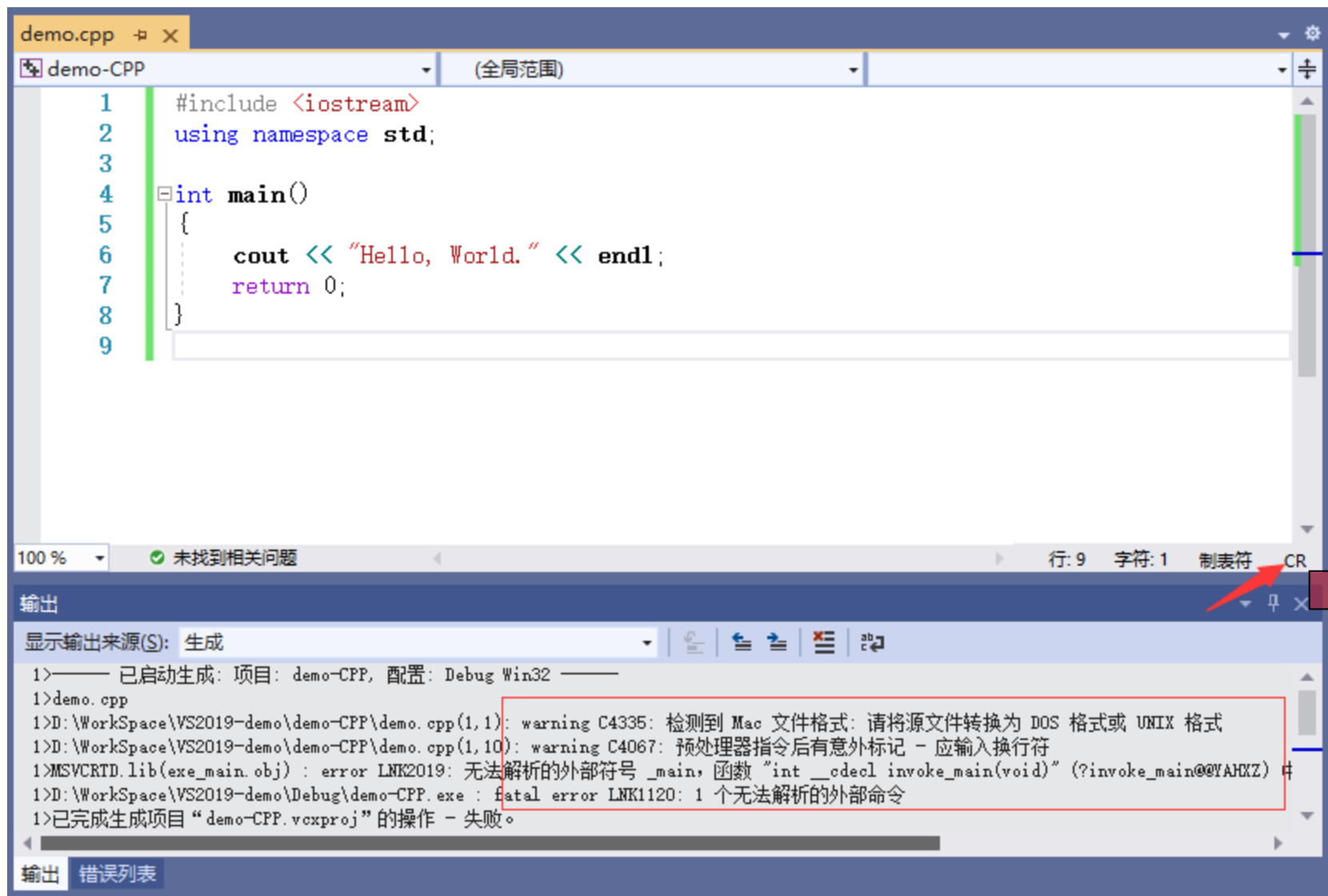
5、不需要填写的部分可以删除 (例如: 某句正确, 则错误部分不填, 或填写“/”即可)



§ 16. 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VS2022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





§ 16. 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 + 复数(2.5)`

编译正确，运行结果：/，能得到此结果的原因：/

编译错误，VS2022下：error C2440: “<function-style-cast>”: 无法从“double”转换为“Complex”

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

错误原因分析：__无复数转double的类型转换函数，也无法理解为double+。没有提供单参数友元函数（即只接收一个double的友元函数），因此复数(2.5)无法匹配任何构造函数。当前类未定义从double到复数的转换，2.5无法自动转换为复数类型__

`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，VS2022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)

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

错误原因分析：__无double+复数的重载。运算符+的参数类型不匹配，友元函数operator+要求两个参数都必须是复数类型或能够隐式转换为复数类型。没有提供单参数构造函数或转换构造函数，2.5不能隐式转换为复数__

`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，VS2022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)

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

错误原因分析：__无double+复数的重载。运算符+的参数类型不匹配，友元函数operator+要求两个参数都必须是复数类型或能够隐式转换为复数类型。没有提供单参数构造函数或转换构造函数，2.5不能隐式转换为复数__

★ 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

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



§ 16. 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 + 复数(2.5)

编译正确，运行结果：/，能得到此结果的原因：/

编译错误，VS2022下：error C2440: “<function-style-cast>”: 无法从“double”转换为“Complex”

Linux下：错误：no matching function for call to 'Complex::Complex(double)'

错误原因分析：__无复数转double的类型转换函数，也无法理解为double+。没有提供单参数构造函数（即只接收一个double的构造函数），因此复数(2.5)无法匹配任何构造函数。当前类未定义从double到复数的转换，2.5无法自动转换为复数类型__

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，VS2022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)

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

错误原因分析：__无double+复数的重载。运算符+的参数类型不匹配，成员函数operator+要求两个参数都必须是复数类型或能够隐式转换为复数类型。没有提供单参数构造函数或转换构造函数，2.5不能隐式转换为复数__

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，VS2022下：错误：no match for 'operator+' (operand types are 'Complex' and 'double')

Linux下：

错误原因分析：__无double+复数的重载。运算符+的参数类型不匹配，成员函数operator+要求两个参数都必须是复数类型或能够隐式转换为复数类型。没有提供单参数构造函数或转换构造函数，2.5不能隐式转换为复数__



§ 16. 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 + 复数(2.5)`

编译正确，运行结果：/，能得到此结果的原因：/

编译错误，VS2022下：error C2440: “<function-style-cast>”：无法从“double”转换为“Complex”

Linux下：错误：no matching function for call to 'Complex::Complex(double)'

错误原因分析：__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+。operator double()定义了将复数对象转换double，但并没有提供double转换为复数机制。因此，2.5被视为double类型，编译器无法直接将它转换为复数类型__

`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，VS2022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)

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

错误原因分析：__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+。编译器将2.5（类型double）隐式转换为复数类型，友元函数operator+被调用，将复数(2.5)和c1相加，返回一个新的复数对象。最终，返回的结果需要赋值给c3，一个复数类型无法被隐式赋值给变量c3__

`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，VS2022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)

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

错误原因分析：__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+。编译器将2.5（类型double）隐式转换为复数类型，友元函数operator+被调用，将复数(2.5)和c1相加，返回一个新的复数对象。最终，返回的结果需要赋值给c3，一个复数类型无法被隐式赋值给变量c3__



§ 16. 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 + 复数(2.5)`

编译正确，运行结果：/，能得到此结果的原因：/

编译错误，VS2022下：error C2440: “<function-style-cast>”：无法从“double”转换为“Complex”

Linux下：错误：no matching function for call to 'Complex::Complex(double)'

错误原因分析：__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+。operator double()定义了将复数对象转换double，但并没有提供double转换为复数机制。因此，2.5被视为double类型，编译器无法直接将它转换为复数类型__

`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，VS2022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)

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

错误原因分析：__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+。编译器将2.5（类型double）隐式转换为复数类型，友元函数operator+被调用，将复数(2.5)和c1相加，返回一个新的复数对象。最终，返回的结果需要赋值给c3，一个复数类型无法被隐式赋值给变量c3__

`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，VS2022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)

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

错误原因分析：__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+。编译器将2.5（类型double）隐式转换为复数类型，友元函数operator+被调用，将复数(2.5)和c1相加，返回一个新的复数对象。最终，返回的结果需要赋值给c3，一个复数类型无法被隐式赋值给变量c3__

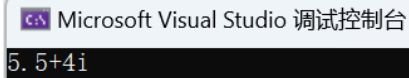


§ 16. 运算符重载 - 转换构造和类型转换函数的使用

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 + 复数(2.5)`

编译正确，运行结果：



，能得到此结果的原因：__通过转换构造函数，double类型可以转换为复数，operator+是一个友元函数，处理两个复数类型的相加__

编译错误，VS2022下：/

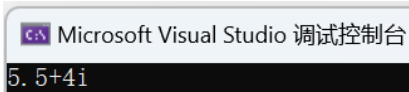
Linux下：/

错误原因分析：/

5.5+4i

`c3 = c1 + 2.5`

编译正确，运行结果：



，能得到此结果的原因：__通过转换构造函数，double类型可以隐式转换为复数，operator+是一个友元函数，处理两个复数类型的相加__

编译错误，VS2022下：/

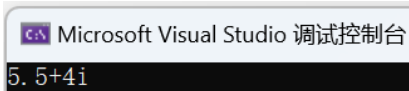
Linux下：/

错误原因分析：/

5.5+4i

`c3 = 2.5 + c1`

编译正确，运行结果：



，能得到此结果的原因：__通过转换构造函数，double类型可以隐式转换为复数，operator+是一个友元函数，处理两个复数类型的相加__

编译错误，VS2022下：/

Linux下：/

错误原因分析：/

5.5+4i

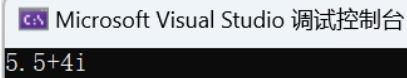


§ 16. 运算符重载 - 转换构造和类型转换函数的使用

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 + 复数(2.5)`

编译正确，运行结果：



，能得到此结果的原因：__通过转换构造函数，double类型可以转换为复数，operator+是一个成员函数，处理两个复数类型的相加__

编译错误，VS2022下：/

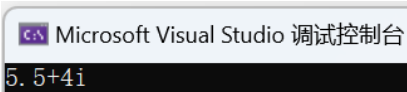
Linux下：/

错误原因分析：/

5.5+4i

`c3 = c1 + 2.5`

编译正确，运行结果：



，能得到此结果的原因：__通过转换构造函数，double类型可以隐式转换为复数，operator+是一个成员函数，处理两个复数类型的相加__

编译错误，VS2022下：/

Linux下：/

错误原因分析：/

5.5+4i

`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，VS2022下：

Linux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)

错误：no match for 'operator+' (operand types are 'double' and 'Complex')

错误原因分析：__成员函数复数 operator+(const 复数 &c2)中，左操作数必须是 复数 类型，因为成员函数的调用方式是c1.operator+(...)。但2.5是double类型，因此不能调用复数的成员函数__

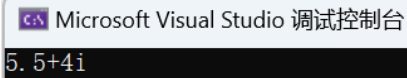


§ 16. 运算符重载 - 转换构造和类型转换函数的使用

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 + 复数(2.5)`

编译正确，运行结果：



，能得到此结果的原因：__通过转换构造函数，double类型可以转换为复数，operator+是一个友元函数，处理两个复数类型的相加__

编译错误，VS2022下：/

Linux下：/

错误原因分析：/

5.5+4i

`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，VS2022下：error C2666: “operator +”：重载函数具有类似的转换

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

错误原因分析：__隐式类型转换导致多个重载都符合，编译器无法选择。编译器会尝试以下两种方式来解析operator+：调用operator+，将2.5转换为复数，并执行复数+复数。使用operator double() 将c1转换为double，然后执行double+double。由于这两种方式都合法且优先级相似，编译器无法决定使用哪种方式，导致重载解析失败__

`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，VS2022下：error C2666: “operator +”：重载函数具有类似的转换

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

错误原因分析：__隐式类型转换导致多个重载都符合，编译器无法选择。编译器会尝试以下两种方式来解析operator+：调用 operator，将2.5转换为复数，并执行复数+复数。使用operator double() 将c1转换为double，然后执行double+double。由于这两种方式都合法且优先级相似，编译器无法决定使用哪种方式，导致重载解析失败__

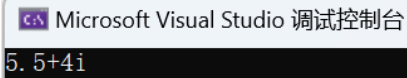


§ 16. 运算符重载 - 转换构造和类型转换函数的使用

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 + 复数(2.5)`

编译正确，运行结果：



，能得到此结果的原因：__通过转换构造函数，double类型可以转换为复数，operator+是一个成员函数，处理两个复数类型的相加__

编译错误，VS2022下：/

Linux下：/

错误原因分析：/

5.5+4i

`c3 = c1 + 2.5`

编译正确，运行结果：/，能得到此结果的原因：/

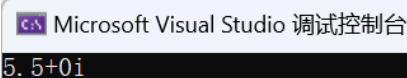
编译错误，VS2022下：error C2666: “Complex::operator +” : 重载函数具有类似的转换

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

错误原因分析：__隐式类型转换导致多个重载都符合，编译器无法选择。编译器会尝试以下两种方式来解析operator+：调用operator+，将2.5转换为复数，并执行复数+复数。使用operator double() 将c1转换为double，然后执行double+double。由于这两种方式都合法且优先级相似，编译器无法决定使用哪种方式，导致重载解析失败__

`c3 = 2.5 + c1`

编译正确，运行结果：



，能得到此结果的原因：__转换构造函数 Complex(double r)，它将double类型的数隐式转换复数类型，但没有为double+复数提供显式的友元重载操作符。因此，编译器选择了转换构造函数，而不是自定义的operator+成员函数，转换构造函数仅复制了double值到real并将imag设置为0__

5.5+0i

编译错误，VS2022下：/

Linux下：/

错误原因分析：/



§ 16. 运算符重载 - 转换构造和类型转换函数的使用

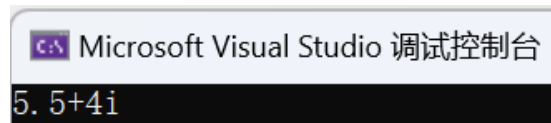
9. 有转换构造函数、有类型转换函数、成员方式实现复数+、另有double+复数

仅讨论语句 `c3 = 2.5 + c1`，回答下列问题

1、为什么编译不错？

在解析 `2.5 + c1` 时，直接调用友元函数计算 `double+复数`。

2、运行结果是多少？



Microsoft Visual Studio 调试控制台
5.5+4i

5.5+4i

3、为什么和8的结果不同？

通过添加友元函数来明确地为表达式 `double+复数` 定义了一个合适的重载。明确告诉编译器，当左侧是 `double` 类型，右侧是复数类型时，直接使用 `friend Complex operator+(const double &d1, const Complex &c2)` 这个重载。C++ 的重载解析规则中，编译器会依次优先考虑：无须类型转换的重载函数；需要单步类型转换的重载函数；其他更复杂的转换路径。



§ 16. 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：__ (a + b)调用成员函数operator+，返回一个复数对象c。因为operator<<没有接收临时对象的重载（接受Complex&），编译器尝试调用类型转换函数operator double()。类型转换函数将c转换为一个double，然后调用标准库中支持的ostream &operator<<(ostream&, double)。最终，打印出的是c的模（即 $\sqrt{\text{real}^2 + \text{imag}^2}$ ），而不是复数的表示。__

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：__friend ostream & operator<<(ostream &out, Complex &c);__
新内容：__friend ostream & operator<<(ostream &out, const Complex &c);__

改动第____36____行，原内容：__ostream &operator<<(ostream &out, Complex &c)__
新内容：__ostream &operator<<(ostream &out, const Complex &c)__