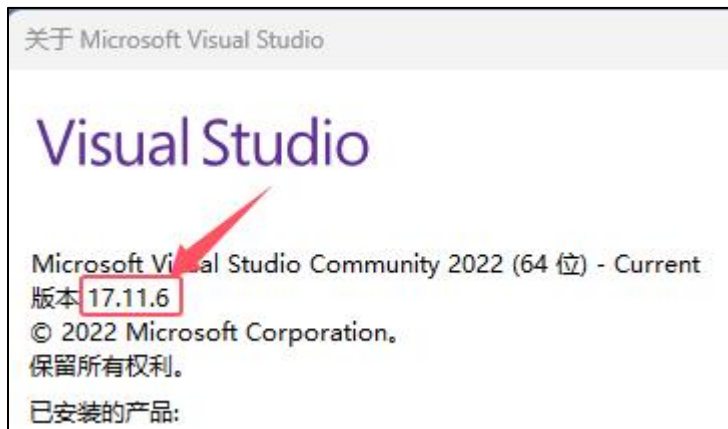


## § 16. 运算符重载 - 转换构造和类型转换函数的使用



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

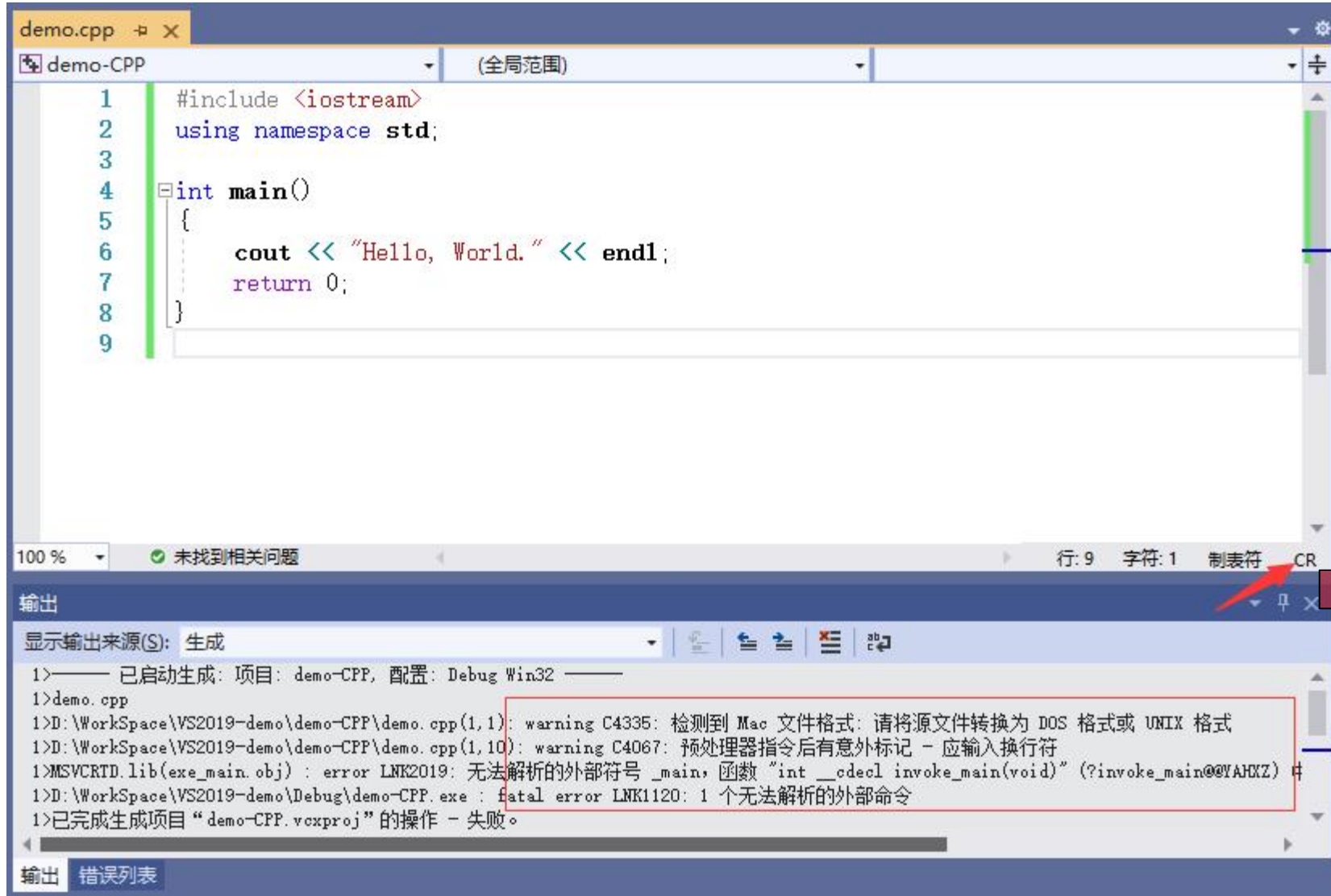
5、不需要填写的部分可以删除（例如：某句正确，则错误部分不填，或填写“/”即可）



## § 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





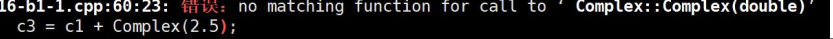
# § 16. 运算符重载 - 转换构造和类型转换函数的使用

## 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 + Complex(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：  （可删除横线后贴图）

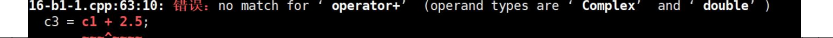
Linux下：  （可删除横线后贴图）

错误原因分析：\_\_\_\_\_Complex的初始化没有从double转来的，无 double转复数 的转换构造函数\_\_\_\_\_

`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，VS2022下：  （可删除横线后贴图）

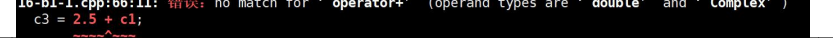
Linux下：  （可删除横线后贴图）

错误原因分析：\_\_\_\_\_没有重载Complex+double的“+”，无 double+复数 的重载\_\_\_\_\_

`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：  （可删除横线后贴图）

Linux下：  （可删除横线后贴图）

错误原因分析：\_\_\_\_\_没有重载double+Complex的“+”，无 double+复数 的重载\_\_\_\_\_

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+

★ 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

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+

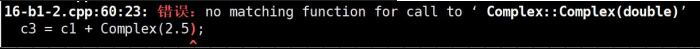
```
c3 = c1 + Complex(2.5)
```

编译正确，运行结果：\_\_\_\_\_能得到此结果的原因：\_\_\_\_\_

编译错误，VS2022下：\_\_\_\_\_（可删除横线后贴图）



Linux下：\_\_\_\_\_（可删除横线后贴图）

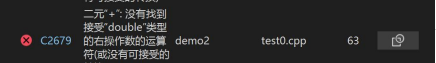


错误原因分析：\_\_\_\_\_Complex的初始化没有从double转来的，无 double转复数 的转换构造函数\_\_\_\_\_

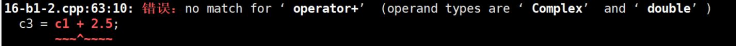
```
c3 = c1 + 2.5
```

编译正确，运行结果：\_\_\_\_\_此结果的原因：\_\_\_\_\_

编译错误，VS2022下：\_\_\_\_\_（可删除横线后贴图）



Linux下：\_\_\_\_\_（可删除横线后贴图）

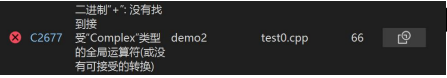


错误原因分析：\_\_\_\_\_没有重载Complex+double的“+”，无 double+复数 的重载\_\_\_\_\_

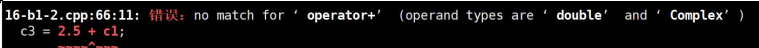
```
c3 = 2.5 + c1
```

编译正确，运行结果：\_\_\_\_\_此结果的原因：\_\_\_\_\_

编译错误，VS2022下：\_\_\_\_\_（可删除横线后贴图）

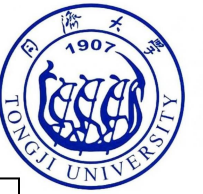


Linux下：\_\_\_\_\_（可删除横线后贴图）



错误原因分析：\_\_\_\_\_没有重载double+Complex的“+”，无 double+复数 的重载\_\_\_\_\_



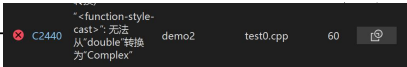


## § 16. 运算符重载 - 转换构造和类型转换函数的使用

### 3. 无转换构造函数、有类型转换函数、友元方式实现复数+

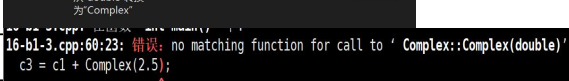
`c3 = c1 + Complex(2.5)`

编译正确，运行结果：



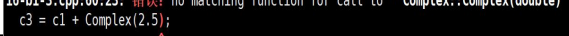
到此结果的原因：\_\_\_\_\_

编译错误，VS2022下：



(可删除横线后贴图)

Linux下：

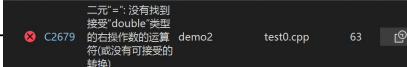


(可删除横线后贴图)

错误原因分析：\_\_\_\_\_Complex的初始化没有从double转来的，无 double转复数 的转换构造函数\_\_\_\_\_

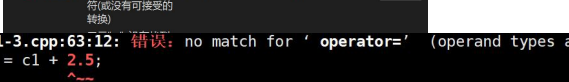
`c3 = c1 + 2.5`

编译正确，运行结果：



到此结果的原因：\_\_\_\_\_

编译错误，VS2022下：



(可删除横线后贴图)

Linux下：

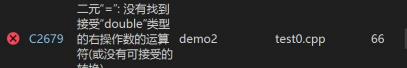


(可删除横线后贴图)

错误原因分析：\_\_\_\_\_右侧的c1由Complex转为了double+2.5之后，由于没有转换构造函数报错\_\_\_\_\_

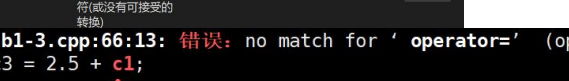
`c3 = 2.5 + c1`

编译正确，运行结果：



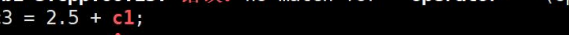
结果的原因：\_\_\_\_\_

编译错误，VS2022下：



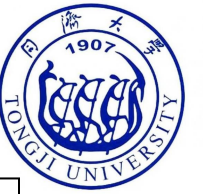
(可删除横线后贴图)

Linux下：



(可删除横线后贴图)

错误原因分析：\_\_\_\_\_右侧的c1由Complex转为了double+2.5之后，由于没有转换构造函数报错\_\_\_\_\_

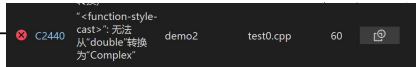


## § 16. 运算符重载 - 转换构造和类型转换函数的使用

### 4. 无转换构造函数、有类型转换函数、成员方式实现复数+

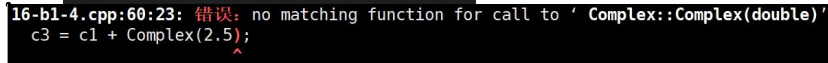
`c3 = c1 + Complex(2.5)`

编译正确，运行结果：



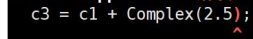
此结果的原因：\_\_\_\_\_

编译错误，VS2022下：



(可删除横线后贴图)

Linux下：

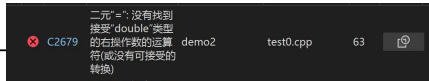


(可删除横线后贴图)

错误原因分析：\_\_\_\_\_Complex的初始化没有从double转来的，无 double转复数 的转换构造函数\_\_\_\_\_

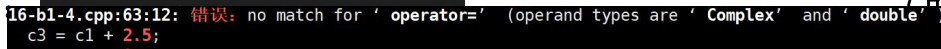
`c3 = c1 + 2.5`

编译正确，运行结果：



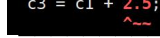
此结果的原因：\_\_\_\_\_

编译错误，VS2022下：



(可删除横线后贴图)

Linux下：

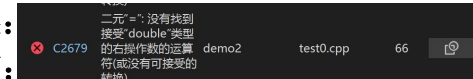


(可删除横线后贴图)

错误原因分析：\_\_\_\_\_右侧的c1由Complex转为了double+2.5之后，由于没有转换构造函数报错\_\_\_\_\_

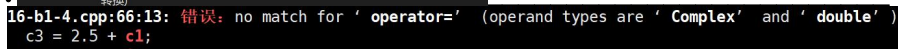
`c3 = 2.5 + c1`

编译正确，运行结果：



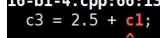
此结果的原因：\_\_\_\_\_

编译错误，VS2022下：



(可删除横线后贴图)

Linux下：



(可删除横线后贴图)

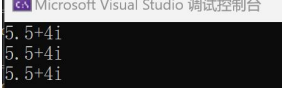
错误原因分析：\_\_\_\_\_右侧的c1由Complex转为了double+2.5之后，由于没有转换构造函数报错\_\_\_\_\_



# § 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 + Complex(2.5)
```

编译正确，运行结果：，能得到此结果的原因： 有double到Complex的转换构造，之后正常相加

编译错误，VS2022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： 有double与Complex的转换构造和重载+，之后正常相加

编译错误，VS2022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
```

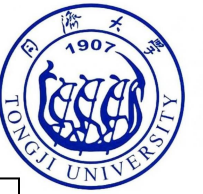
编译正确，运行结果：，能得到此结果的原因： 有double与Complex的转换构造和重载+，之后正常相加

编译错误，VS2022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： \_\_\_\_\_

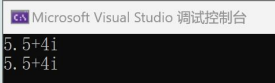




# § 16. 运算符重载 - 转换构造和类型转换函数的使用

## 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 + Complex(2.5)
```

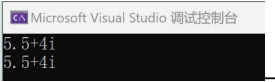
编译正确，运行结果：，能得到此结果的原因：\_\_\_有double到Complex的转换构造，之后正常相加\_\_\_

编译错误，VS2022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：\_\_\_有double与Complex的转换构造和重载+，之后正常相加\_\_\_

编译错误，VS2022下：\_\_\_\_\_（可删除横线后贴图）

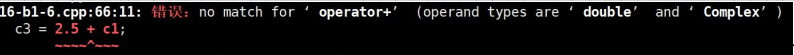
Linux下：\_\_\_\_\_（可删除横线后贴图）

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，VS2022下：\_\_\_\_\_（可删除横线后贴图）

Linux下：\_\_\_\_\_（可删除横线后贴图）

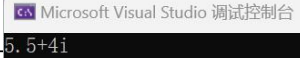
错误原因分析：\_\_\_\_\_c2 + c1 的操作会被翻译为 c2.operator+(c1)，即调用 c2 的成员函数，而不是 c1 的成员函数



## § 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 + Complex(2.5)`

编译正确，运行结果：能得到此结果的原因：\_\_\_\_有double到Complex的转换构造，之后正常相加\_\_\_\_


编译错误，VS2022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：\_\_\_\_

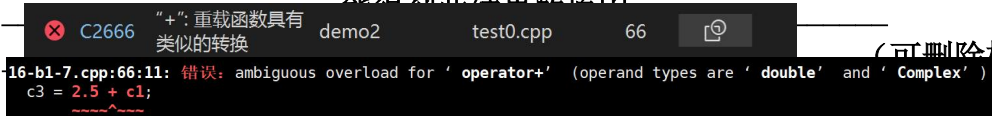
编译错误，VS2022下：（可删除横线后贴图）

Linux下：\_\_\_\_（可删除横线后贴图）

错误原因分析：\_\_\_\_友元函数造成的重载二义性\_\_\_\_

`c3 = 2.5 + c1`

编译正确，运行结果：\_\_\_\_

编译错误，VS2022下：（可删除横线后贴图）

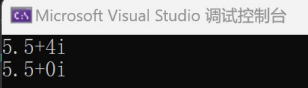
Linux下：\_\_\_\_（可删除横线后贴图）

错误原因分析：\_\_\_\_友元函数造成的重载二义性\_\_\_\_



# § 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 + Complex(2.5);`  能得到此结果的原因: 有double到Complex的转换构造, 之后正常相加

编译正确, 运行结果: 5.5+4i  
5.5+0i

编译错误, VS2022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: \_\_\_\_\_

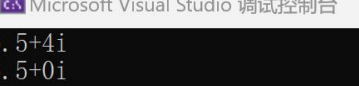
`c3 = c1 + 2.5;`

编译正确, 运行结果:  能得到此结果的原因: \_\_\_\_\_

编译错误, VS2022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: 成员函数负责c1 Complex型后面的operator+, 和类型转换函数冲突

`c3 = 2.5 + c1;`  得到此结果的原因: 成员函数不能负责2.5double型后面的operator+, 因此类型转换控制了c1转化为double

编译正确, 运行结果: 5.5+4i  
5.5+0i

编译错误, VS2022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: \_\_\_\_\_



## § 16. 运算符重载 - 转换构造和类型转换函数的使用

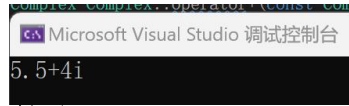
9. 有转换构造函数、有类型转换函数、成员方式实现复数+、另有double+Complex

仅讨论语句  $c3 = 2.5 + c1$ ，回答下列问题

1、为什么编译不错？

特定了double+Complex的计算方法

2、运行结果是多少？



3、为什么和8的结果不同？

8未确定所以会出现冲突，而9指定了出现double+Complex会如何计算



## § 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：\_\_\_但是，Complex 类型的结果被传递到 cout 时，ostream& operator<<(ostream&, Complex&) 没有被调用，而是调用了 类型转换函数 operator double()\_\_\_

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)\_\_\_

```
Microsoft Visual Studio 调试控制台
1+1i
2+2i
3+3i
4. 24264
```