

```
1 Exercise. Python Class_answer
2
3 1. 복소수 class를 정의하기
4   -real(실수)와 img(허수)를 member variable로
5   -Classname : MyComplex
6
7 2. 덧셈, 뺄셈, 곱셈 연산자 지원
8   -__add__()
9   -__sub__()
10  -__mul__()
11  -복소수의 곱일 경우 복소수와 실수를 곱할 때와 복소수와 복소수를 곱할 때 모두 처리가능해야 함.
12
13 3. 길이(복소수의 크기) 지원
14  -__len__()
15  -반드시 정수를 반환해야 함.
16
17 4. 복소수 출력 '1 + 4j'와 같이 표현
18  -__str__()
19  -허수부가 음수일 경우에는 a - bj로 표현해야 함.
20
21 5. 비교 연산 ==, != 지원
22  -__eq__()
23  -__ne__()
24  -True,False 반환
25
26 6. 복소수의 절대값 구하기
27  -math module의 sqrt() 사용
28
29 7. Test
30  su = MyComplex(3,6)
31  print(su)
32
33  su1 = MyComplex(7, 9)
34  print(su + su1)
35  print(su - su1)
36
37  print(su * 3)
38  print(su * su1)
39
40  print(su == su1)
41  print(su != su1)
42
43  print(abs(su))
44  print(len(su))
45
46
47 -----
48 import math
49
50 class MyComplex :
51     def __init__(self, real, img):
```

```
52     self.real = real
53     self.img = img
54
55     def __str__(self):
56         if self.img >= 0:
57             return '{} + {}j'.format(self.real, self.img)
58         elif self.img < 0 :
59             return '{} - {}j'.format(self.real, self.img * -1)
60
61     def __add__(self, othercomplex):
62         return MyComplex(self.real + othercomplex.real, self.img + othercomplex.img)
63
64     def __sub__(self, othercomplex):
65         return MyComplex(self.real - othercomplex.real, self.img - othercomplex.img)
66
67     def __mul__(self, x):
68         if type(x) == int:
69             return MyComplex(self.real * x, self.img * x)
70         elif type(x) == MyComplex:
71             #(a + bj) x (c + dj) -> (ac - bd) + (ad + bc)j
72             return MyComplex(self.real * x.real - self.img * x.img, self.real * x.img + self.img *
73                               x.real)
74
75     def __eq__(self, othercomplex):    #==
76         return self.real == othercomplex.real and self.img == othercomplex.img
77
78     def __ne__(self, othercomplex):    #!=
79         return not (self.real == othercomplex.real and self.img == othercomplex.img)
80
81     def __abs__(self):
82         return math.sqrt(self.real ** 2 + self.img ** 2)
83
84     def __len__(self):
85         return int(math.sqrt(self.real ** 2 + self.img ** 2))
86
87 su = MyComplex(3,6)
88 print(su)
89
90 su1 = MyComplex(7, 9)
91 print(su + su1)
92 print(su - su1)
93
94 print(su * 3)
95 print(su * su1)
96
97 print(su == su1)
98 print(su != su1)
99
100 print(abs(su))
101 print(len(su))
```