

```

1 Lab. Python Class Basic II
2
3 1. 사용 tool
4 -Jupyter Notebook
5 -Microsoft Visual Studio Code
6
7 2. Code
8
9 class Point:
10     def __init__(self, x, y):
11         self.x = x
12         self.y = y
13
14     def print_pt(self):
15         print(f'({self.x}, {self.y})')
16
17     def add(self, pt):
18         new_x = self.x + pt.x
19         new_y = self.y + pt.y
20         return Point(new_x, new_y)
21
22     def multiply(self, factor):
23         return Point(self.x * factor, self.y * factor)
24
25     def length(self):
26         return self.x ** 2 + self.y ** 2
27
28     def get_x(self):
29         return self.x
30
31     def get_y(self):
32         return self.y
33
34     #Base Overloading Methods
35     def __str__(self):
36         return f'({self.x}, {self.y})'
37
38     def __add__(self, pt):
39         new_x = self.x + pt.x
40         new_y = self.y + pt.y
41         return Point(new_x, new_y)
42
43     def __sub__(self, pt):
44         new_x = self.x - pt.x
45         new_y = self.y - pt.y
46         return Point(new_x, new_y)
47
48     def __mul__(self, factor):
49         return Point(self.x * factor, self.y * factor)
50
51     def __len__(self):
52         return self.x ** 2 + self.y ** 2
53
54     def __getitem__(self, index):
55         if index == 0 : return self.x
56         elif index == 1 : return self.y
57
58
59
60 p1 = Point(100, 200)
61 p2 = Point(300, 450)
62
63 p1.print_pt() # (100, 200)
64 p2.print_pt() # (300, 450)
65
66 print(p1) # (100, 200) call __str__()
67 print(p2) # (300, 450)
68
69 p3 = p1.add(p2)
70 print(p3) # (400, 650)
71
72 p4 = p1 + p2 # call __add__()
73 print(p4) # (400, 650)
74
75 p5 = p2 - p1 # call __sub__()
76 print(p5) # (200, 250)
77
78 p6 = p1.multiply(7)
79 print(p6) # (700, 1400)
80
81 p7 = p1 * 7
82 print(p7) # (700, 1400)
83
84

```

```
85 print(f'p1\'s length = {p1.length()}') #p1's length = 50000
86 print(f'p1\'s length = {len(p1)}') #call __len__() p1's length = 50000
87
88 print(f'p1(x, y) = ({p1.get_x()}, {p1.get_y()}') #p1(x, y) = (100, 200)
89 print(f'p1(x, y) = ({p1[0]}, {p1[1]})') #p1(x, y) = (100, 200)
90
91
92 Refer to 3.3.8. Emulating numeric
types(https://docs.python.org/3/reference/datamodel.html?object.add #emulating-numeric-types)
```