

(17-B) Simple OOP Practice

Simple OOP Project (1): Animal and Dog

```
class Animal:

    def __init__(self):
        print("Animal created")

    def whoAmI(self):
        print("Animal")

    def eat(self):
        print("Eating")

class Dog(Animal):

    def __init__(self):
        super().__init__()

        print("Dog created")

    def whoAmI(self):
        print("Dog")

    def bark(self):
        print("Woof!")
```

왼쪽 code를 설명하고 아래 code 결과를 보이시오

정답

Animal이라는 Class는 parameter로 아무것도 받지 않으며, Class가 생성됨과 동시에 "Animal created"라는 string을 출력하고, whoAmI라는 메소드를 실행하면, "Animal"이라는 string을 출력한다(내가 누구냐고 물어보면 동물이다 라고 출력함). 또, eat이라는 메소드를 실행하면, "Eating"이라는 string을 출력하게 한다(먹는 메소드를 실행하면 먹는중이라고 출력함).

Class Dog는 Class Animal 의 sub-class로, parameter로 아무것도 받지 않는다. 이 때, super()를 사용하여 부모 클래스인 Animal의 __init__ 메소드를 호출함과 동시에, "Dog created"라는 string도 출력한다.

whoAmI라는 instance function이 Dog class 내에서는 "Dog"라는 string을 출력하게 한다(누구냐고 물어보면 동물이다라고 출력하는 것이 아니라 이제 개라고 출력함).

Bark라는 메소드는 Dog class 내에서 "Woof!"라는 string을 출력시킨다(짖는 메소드에서는 짖는 소리를 출력함).

정답

```
d = Dog()
d.whoAmI()
d.eat()
d.bark()
```

```
Animal created
Dog created
Dog
Eating
Woof!
```

Simple OOP Project (2): Circle

- Implement Circle Class with the following specification
 - The value of π is 3.141592
 - The `area()` method returns the area of a circle
 - The `setRadius()` method sets a new value for the radius
 - The `getRadius()` method gets the current radius
- The usage of Circle Class is as follows

```
c = Circle()

c.setRadius(5)
print(c.getRadius())
print(c.area())
```

정답

```
class Circle():

    def __init__(self):
        self.pi = 3.141592
        self.radius = 0

    def setRadius(self, r):
        self.radius = r

    def getRadius(self):
        return self.radius

    def area(self):
        return self.pi * self.radius**2
```

Simple OOP Project (3): Shape and Others

```
#An example of a class
class Shape:

    def __init__(self, x, y):
        self.x = x
        self.y = y
        self.description = "This shape has not been described yet"
        self.author = "Nobody has claimed to make this shape yet"

    def area(self):
        return self.x * self.y

    def perimeter(self):
        return 2 * self.x + 2 * self.y

    def describe(self, text):
        self.description = text

    def authorName(self, text):
        self.author = text

    def scaleSize(self, scale):
        self.x = self.x * scale
        self.y = self.y * scale
```

- 위의 code를 설명하시오

정답

Class Shape에서 parameter로 x, y를 받고, 인스턴스변수 self.x에는 초기값으로 메소드 내에서 받은 x값을, 인스턴스변수 self.y에는 초기값으로 메소드 내에서 받은 y값을 할당해준다. Class가 받는 parameter가 x와 y 두 개이므로, 아마도 Shape는 직사각형이거나 변수가 두 개인것 만으로도 정의되는 도형이어야 한다. 인스턴스 변수 self.description과 self.author의 초기값을 보면 알 수 있듯이, 어떤 모양인지와 누가 모양을 만들라 했는지에 대한 부분은 받지 않았으므로, class내부의 메소드에서 받을 것임을 예상할 수 있다.

area라는 메소드는 면적을 구하는 메소드로, 변수 self.x와 변수 self.y를 곱해준 값, 즉 직사각형의 면적을 반환한다.

perimeter라는 메소드는 직사각형의 둘레를 구하는 메소드로, 두 변수 self.x와 self.y에 두 배씩 하여 더한 값, 즉 둘레를 반환한다.

describe라는 메소드는 text라는 parameter를 받아 class에서 지정한 변수 self.description을 메소드 내에서 받은 parameter인 text로 교체한다.

authorName 메소드는 text라는 parameter를 받아 class에서 지정한 변수 self.author를 메소드 내에서 받은 parameter인 text로 교체한다.

scaleSize라는 메소드는 scale이라는 parameter를 받아 class에서 초기값을 x, y로 지정해준 변수 self.x, self.y를 scale만큼 곱하여 재 지정한다. (즉, scale 만큼 곱하여 직사각형의 변의 길이를 조작한다.)

Simple OOP Project (3): Shape and Others [1/3]

아래 code의 수행결과를 쓰시오

```
rectangle = Shape(100, 45)
#finding the area of your rectangle:
print(rectangle.area())

#finding the perimeter of your rectangle:
print(rectangle.perimeter())

#describing the rectangle
rectangle.describe("A wide rectangle, more than twice\
as wide as it is tall")

#making the rectangle 50% smaller
rectangle.scaleSize(0.5)

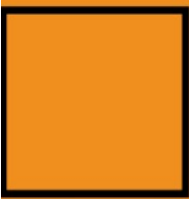
#re-printing the new area of the rectangle
print(rectangle.area())
```

정답

```
4500
290
1125.0
```

Simple OOP Project (3): Shape and Others [2/2]

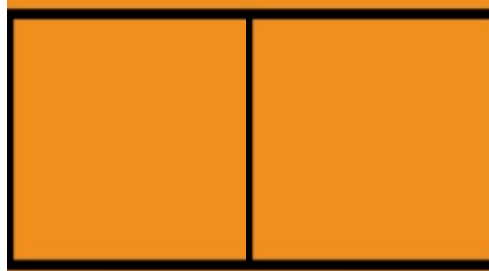
- Square class를 Shape class를 이용하여 만드시오.



정답

```
class Square(Shape):  
    def __init__(self, a):  
        super().__init__(a, a)
```

- Double Square class를 Square class를 이용하여 만드시오

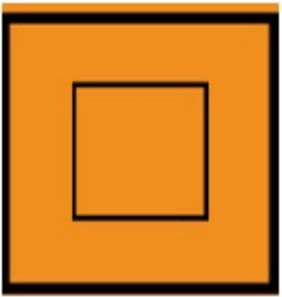


정답

```
class DoubleSquare(Square):  
    def perimeter(self):  
        return 4 * self.x + 2 * self.y  
  
    def area(self):  
        return 2 * self.x * self.y
```

Simple OOP Project (3): Shape and Others

- Inside Double Square class를 Square class를 이용하여 만드시오
- 내부에 있는 square는 외부 square의 1/4 면적을 가진다고 가정하시오



정답

```
class InsideDoubleSquare(Square):  
  
    def perimeter(self):  
        return self.x + self.y  
  
    def area(self):  
        return self.x * self.y / 4.0
```