

# 연습문제 : 객체와 클래스

## 1. 파이썬의 클래스와 객체 지향 프로그래밍에 대한 설명으로 틀린 것은?

- ① 클래스에서 상속은 부모 클래스로부터 속성과 메서드를 물려받은 자식 클래스를 생성하는 것을 말한다.
- ② 클래스에서 `__init__` ( ) 함수는 객체 초기화 예약 함수이다.
- ③ 객체 지향 프로그래밍에서 속성은 값(variable)으로, 행동은 메서드(method)로 표현된다.
- ④ 클래스에서 함수(function) 추가는 기존 함수의 사용법과 동일하다.
- ⑤ 다형성은 같은 이름의 메서드의 내부 로직을 다르게 작성하는 것을 말하며, 같은 부모 클래스를 상속하는 과정에서 주로 발생한다.

## 2. 다음 코드의 실행 결과를 쓰시오.

```
class Person(object):
    def __init__(self, name):
        self.name = name

    def language(self):
        pass

class Earthling(Person):
    def language(self, language):
        return language

class Groot(Person):
    def language(self, language):
        return "I'm Groot!"

name = ['Gachon', 'Dr.Strange', 'Groot']
country = ['Korea', 'USA', 'Galaxy']
language = ['Korean', 'English', 'Groot']

for idx, name in enumerate(name):
    if country[idx].upper() != 'GALAXY':
        person = Earthling(name)
        print(person.language(language[idx]))

    else:
        groot = Groot(name)
        print(groot.language(language[idx]))
```

3. 다음 코드의 실행 결과를 쓰시오.

```
class SoccerPlayer(object):
    def __init__(self, name, position, back_number):
        self.name = name
        self.position = position
        self.back_number = back_number

    def change_back_number(self, back_number):
        self.back_number = back_number

jinhyun = SoccerPlayer("jinhyun", "MF", 10)
print("현재 선수의 등 번호는:", jinhyun.back_number)
jinhyun.change_back_number(5)
print("현재 선수의 등 번호는:", jinhyun.back_number)
```

4. 다음과 같은 코드는 객체 지향 프로그램의 어떤 특징을 보여 주는지 모두 고르시오.

```
class Class(object):
    def __init__(self, name, score):
        self.name = name
        self.score = score

class Math(Class):
    def say( ):
        print("힘내")
```

- ① 다형성                      ② 가시성                      ③ 상속  
④ 인스턴스                      ⑤ 속성

5. 다음 (가)와 (나)에 들어갈 알맞은 단어를 고르시오.



## 6. 다음 코드의 실행 결과를 쓰시오

```
class Marvel(object):
    def __init__(self, name, characteristic):
        self.name = name
        self.characteristic = characteristic
    def __str__(self):
        return "My name is {0} and my weapon is {1}.".format(self.name, self.characteristic)

class Villain(Marvel):
    pass

first_villain = Villain("Thanos", "infinity gauntlet")
print(first_villain)
```

## 7. 제시된 코드를 참고하여 질문에 답하라.

```
class TV(object):
    def __init__(self, size, year, company):
        self.size = size
        self.year = year
        self.company = company
    def describe(self):
        print(self.company + "에서 만든 " + self.year + "년형 " + self.size + "인치" + "TV")

class Laptop(TV):
    def describe(self):
        print(self.company + "에서 만든 " + self.year + "년형 " + self.size + "인치 " + "노트북")

LG_TV = TV("32", "2019", "LG")
LG_TV.describe()

samsung_microwave = Laptop("15", "2018", "Samsung")
samsung_microwave.describe()
```

(가) 다음 코드의 출력 결과를 차례대로 쓰시오.

(나) 다음은 TV 클래스와 Laptop 클래스의 관계에 대한 설명이다. 빈칸에 알맞은 단어를 넣으시오.

위 코드에서 Laptop 클래스는 TV 클래스를 (     가     )했습니다. 또한, 같은 이름의 내부 로직을 다르게 작성했으므로 (     나     )의 사례로도 볼 수 있습니다.

8. 제시된 코드를 참고하여 질문에 답하라.

```
class Person:
    def __init__(self, name, age, position):
        self.Name = name
        self.Age = age
        self.Position = position
    def show_info(self):
        print('이름 : {}'.format(self.Name))
        print('나이 : {}'.format(self.Age))
        print('직위 : {}'.format(self.Position))
        print("저는 {} {}입니다. 나이는 {}입니다.".format(self.Position, self.Name, self.Age))

class Researcher(Person):
    def __init__(self, name, age, position, degree):
        Person.__init__(self, name, age, position)
        self.Degree = degree
    def show_info(self):
        Person.show_info(self)
        print("저는 {} 입니다.".format(self.Degree))

if __name__ == '__main__':
    researcher_john = Researcher("John", "22", "연구원", "학사")
    researcher_tedd = Researcher("Tedd", "40", "소장", "박사")
    researcher_john.show_info()
    print("="*50)
    researcher_tedd.show_info()
```

(가) 다음 코드의 출력 결과를 차례대로 쓰시오.

(나) 문제의 Person 클래스와 Researcher 클래스는 파이썬 클래스의 어떤 속성을 보여주는가?

## 9. 다음 코드의 실행 결과를 쓰시오.

```
class Score:
    def __init__(self,student):
        tmp = student.split(",")
        self.name = tmp[0]
        self.midterm = int(tmp[1])
        self.final = int(tmp[2])
        self.assignment = int(tmp[3])
        self.score = None
        self.grade = None

    def total_score(self):
        test_score = ((self.midterm + self.final)/2)*0.8

        if self.assignment>=3:
            assign_score = 20
        elif self.assignment>=2:
            assign_score = 10
        elif self.assignment>=1:
            assign_score = 5
        else:
            assign_score = 0

        self.score = test_score + assign_score

    def total_grade(self):
        if self.assignment==0:
            grade = "F"
        elif self.score >=90:
            grade = "A"
        elif self.score >=70:
            grade = "B"
        elif self.score >=60:
            grade = "C"

        else:
            grade = "F"

        self.grade = grade
        return grade

student_john = Score("john,90,90,0")
aa = student_john.total_score()
bb = student_john.total_grade()
print(aa,bb,student_john.score,student_john.grade)
```

10. 다음 코드의 실행 결과를 쓰시오.

```
class IceCream(object):
    def __init__(self, flavor):
        self.flavor = flavor
    def change_flavor(self, new_flavor):
        print('아이스크림을 %s에서 %s로 변경해주세요.' %(self.flavor, new_flavor))
        self.flavor = new_flavor
        print('아이스크림 맛을 %s로 변경해드렸어요.' %self.flavor)

ice_cream = IceCream('레인보우 샤베트')
ice_cream.change_flavor('바람과 함께 사라지다')
```

11. 다음 코드의 실행 결과를 쓰시오.

```
class Terran(object):
    def __init__(self, mineral):
        self.scv = 4
        self.marine = 0
        self.medic = 0
        self.mineral = mineral
    def command(self, SCV=False):
        self.mineral += 8*self.scv
        if SCV:
            self.scv += 1
            self.mineral -= 10
    def barrack(self, Marine=False, Medic=False):
        self.mineral += 8*self.scv
        if Marine:
            self.marine += 1
            self.mineral -= 15
        if Medic:
            self.medic += 1
            self.mineral -= 25
    def check_source(self):
        print("Mineral: "+str(self.mineral))

User = Terran(50)
User.command(True)
User.barrack(True,True)
User.check_source()
```