

- 01** 객체 지향 프로그래밍 Object Oriented Programming, OOP은 함수처럼 어떤 기능을 함수 코드에 묶어 두는 것이 아니라, 그런 기능을 묶은 하나의 단일 프로그램을 객체라고 하는 코드에 넣어 다른 프로그래머가 재사용할 수 있도록 하는, 컴퓨터 공학의 오래된 프로그래밍 기법 중 하나이다.
- 02** 객체 object는 실생활에 존재하는 실제적인 물건 또는 개념을 뜻하며, 속성 attribute과 행동 action으로 구성된다. 객체 지향 프로그래밍은 이러한 객체의 개념을 활용하여 프로그램으로 표현하는 기법이다. 속성은 변수 variable로, 행동은 함수 function로 정의된다.
- 03** 클래스 class는 객체가 가져야 할 기본 정보를 담은 코드로, 일종의 설계도 코드라고 생각하면 된다.
- 04** 파이썬에서 클래스를 선언하기 위한 기본 코드 템플릿은 다음과 같다.

```
class SoccerPlayer(object):
```

클래스 예약어 클래스 이름 상속받는 객체명

- 05** `__init__()` 함수는 class에서 사용할 변수를 정의하는 함수이다.
- 06** 파이썬에서 `_`의 쓰임은 개수에 따라 여러 가지로 나눌 수 있다.
- 07** 인스턴스 instance는 클래스에서 실제적인 데이터가 입력되어 사용할 수 있는 형태의 객체를 뜻한다.
- 08** 클래스를 사용하는 이유에는 여러 가지가 있지만, 핵심은 자신의 코드를 다른 사람이 손쉽게 사용할 수 있도록 설계하기 위함이다.

09 객체 지향 프로그래밍의 특징에는 크게 상속, 다형성, 가시성이 있다.

10 상속inheritance은 이름 그대로 무엇인가를 내려받는 것을 뜻한다. 좀 더 자세히 설명하면 부모 클래스에 정의된 속성과 메서드를 자식 클래스가 물려받아 사용하는 것이다.

11 다형성polymorphism은 같은 이름의 메서드가 다른 기능을 할 수 있도록 하는 것을 말한다.

12 가시성Visibility은 객체의 정보를 볼 수 있는 레벨을 조절하여 객체의 정보 접근을 숨기는 것을 말하며, 다양한 이름으로 불린다. 파이썬에서는 가시성이라고 하지만, 좀 더 중요한 핵심 개념은 캡슐화encapsulation와 정보 은닉information hiding이다.

01 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.

02 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]))

```

03 다음과 같은 2개의 파일이 있다. 'main.py'를 실행하였을 때 나오는 결과로 알맞은 것은?

factorial_calculator.py

```

def factorial(n):
    if n == 0:
        return 1
    else:
        return (n * factorial(n-1))

```

main.py

```

from factorial_calculator import factorial
print(factorial_calculator.factorial(6))

```

- | | | |
|-------------|-------------|-------|
| ① NameError | ② None | ③ 120 |
| ④ 720 | ⑤ TypeError | |

04 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 등번호는:", jinhyun.back_number)
jinhyun.change_back_number(5)
print("현재 선수의 등번호는:", jinhyun.back_number)
```

05 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

06 다음 (가)와 (나)에 들어갈 알맞은 단어를 쓰시오.



붕어빵틀
(가)



붕어빵
(나)

(가) _____

(나) _____

07 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)
```

08 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()
```

(1) 코드의 출력 결과를 차례대로 쓰시오.

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

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

(가) _____ (나) _____

09 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 가천대학교 연구소 {0} {1}입니다. 나이는 {2}입니
다.".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()
```

(1) 코드의 출력 결과를 차례대로 쓰시오.

(2) Person 클래스와 Researcher 클래스는 파이썬 클래스의 어떤 속성을 보여 주는가?

10 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)

```

11 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')

```

12 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()
```