웹파이썬프로그래밍**01** 기말고사

2022. Fall

- 본인의 파일 이름이 **_**학번**.py** 형식을 갖추고 있는지 확인 → 틀리면 -1점
- 불필요한 코드는 모두 삭제 → 함수, 클래스 이외의 다른 코드에서 에러 발생시 -1점
- 불필요한 print() 주석처리 또는 삭제
- 제출을 완료 후, 시험 시간이 종료될 때까지 노트북을 덮고
  책이나 프린트물로 다른 시험 공부 가능 (디지털 기기 사용 금지)

<시험에 필요한 정보**>**

[Tip] *str.upper()* returns a string where all characters are in uppercase.
[Tip] *str.lower()* returns a string where all characters are in lowercase.
[Tip] use help() if necessary (press 'q' button to quit from help() menial on the VS Code)
[Tip] press Ctrl+C (or Command+C on Mac) to escape from the infinite loop on VS Code.

[Problem #1] (3 points)

- The function name: **function01**
- This function takes one argument
  - 오직 int type의 값만 요소로 갖는 tuple 또는 list
- The return value
  - 주어진 list 또는 tuple의 요소 중 홀수의 합을 반환. 아무 요소가 없다면 0 반환
- 함수는 아래와 같이 keyword argument를 사용하여 테스트
  - function01(S=[1, 2, 3, 4, 5]) returns 9
  - function01(S=(1, 3, 2)) returns 4
  - function01(S=[]) returns 0
  - function01(S=[-2, 0, 1]) returns 1

[Problem #2] (3 points)

- The function name: **function02**
- This function takes one argument
  - str type (두 글자 이상이며, 알파벳으로만 구성됨. 공백 없음)
- The return value (bool type)
  - 대소문자 상관없이, 주어진 문자열이 팰린드롬(palindrome)이라면 True,
    팰린드롬이 아니라면 False를 반환
  - [Tip] 팰린드롬이란 "우영우", "기러기", "인도인", "level", "mygym", "ABCcba"
    등과 같이 거꾸로 뒤집어도 원래의 단어와 동일한 문자열을 의미.
- 함수는 아래와 같이 positional arguments를 사용하여 테스트
  - function02("level") returns True
  - function02("levEL") returns True

    - function02(“Aa”) returns True
    - function02(“hello”) returns False (because “hello” != “olleh”)
    - function02(“pythoN”) return False (because “pythoN” != “Nohtyp”)

[Problem #3] (3 points)

- The function name: **function03**
- This function takes the unknown number of positional arguments
    - 몇 개의 인자가 넘어올지 알 수 없지만, 모든 인자는 int type
- The return value
    - 모든 인자 중 짝수의 합을 계산하여 반환. 아무 인자가 없다면 0 반환
- 함수는 아래와 같이 positional arguments를 사용하여 테스트
    - function03() returns 0
    - function03(1) returns 0
    - function03(1, 2, 3) returns 2
    - function03(1, -2, 3, 6) returns 4

[Problem #4] (3 points)

- The function name: **function04**
- This function takes two arguments
    - first argument: int type
    - second argument: int type or str type
- The return value (float or Nonetype)
    - 첫 번째 인자 값을 두 번째 인자로 나눈 값을 반환
    - 단, 혹시 어떤 에러가 발생한다면 무조건 None을 반환
- 함수는 아래와 같이 positional arguments를 사용하여 테스트
    - function04(1, “10”) returns None
    - function04(1, 10) returns 0.1
    - function04(0, 0) returns None
    - function04(99, 99) returns 1.0

[Problem #5] (4 points)

- Please define **Person** class so that you can see the [Result] when running the [Code].
- **Person** class has two member variables
    - name: str type
    - height: int type

[Code] → I will test your class using this code using various name and height values

```
p = Person()
print("p.name:", p.name)
print("p.height:", p.height)
```

```
p.change_name("Kim")
p.change_height(170)
print("p.name:", p.name)
print("p.height:", p.height)
```

[Result]

```
p.name: Park
p.height: 160
p.name: Kim
p.height: 170
```

[Problem #6] (4 points)

- Please define **Animal** and **Dog** class so that you can see the [Result] when running the [Code].

[Code] → I will test your class using this code

```
a = Animal()
d = Dog()
if issubclass(Dog, Animal):
    print(a.cry())
    print(d.cry())
```

[Result]

```
Cry
Woof
```

[Problem #7] (5 points)

- Please define **Human** class so that you can see the [Result] when running the [Code]
- **Human** class has two member variables
    - name: str type
    - age: int type

[Code] → I will test your class using this code

```
h1 = Human()
h2 = Human("Kim", 30)
print("h1.name:", h1.name)
print("h1.age:", h1.age)
print("h2.name:", h2.name)
```

```
print("h2.age:", h2.age)
```

[Result]

```
h1.name: Park
h1.age: 20
h2.name: Kim
h2.age: 30
```

[Problem #8] (5 points)

- Please define **Number** class so that you can see the [Result] when running the [Code].
- **Number** class has one member variable
    - number: int type or float type

[Code] → I will test your class using the following code with various int or float values.

```
n1 = Number()
n2 = Number(3.15)
print(n1 + n2)  # n1 + n2 returns int or float type value
print(n1 - n2)  # n1 - n2 returns int or float type value
print("n1.number:", n1.number)
print("n2.number:", n2.number)
```

[Result]

```
4.15
-2.15
n1.number: 1
n2.number: 3.15
```

[Problem #9] (5 points)

- Please define **Student** class so that you can see the [Result] when running the [Code].
- **Student** class has one member variable
    - student_number: int type
- == compares student_numbers of the instances and returns True or False (bool type)

[Code] → I will test your class using the following code with various int values.

```
s1 = Student(1234)
s2 = Student(5678)
s3 = Student(1234)
print(s1==s1)  # s1==s1 returns bool type value, True or False
print(s1==s2)  # s1==s2 returns bool type value, True or False
```

```
print(s1==s3)  # s1==s2 returns bool type value, True or False
print(s1.student_number)
print(s2.student_number)
print(s3.student_number)
```

[Result]

```
True
False
True
1234
5678
1234
```

[Problem #10] (5 points)

- Please define **Account** class so that you can see the [Result] when running the [Code].
- **Account** class has one member variable
  - balance: int type

[Code] → I will test your class using the following code with various int values.

```
a1 = Account(0)  # initial money
a2 = Account(100)  # initial money
print("a1's balance:", a1.balance)
print("a2's balance:", a2.balance)
a1.deposit(20)  # 입금
a2.withdraw(10)  # 출금
print("a1's balance:", a1.balance)
print("a2's balance:", a2.balance)
```

[Result]

```
a1's balance: 0
a2's balance: 100
a1's balance: 20
a2's balance: 90
```

- THE END -

Web/Python Programming - Final exam

2022. Fall

Checklist

- Check if your file name is **_STUDENTNUMBER.py** → -1 point if it is wrong
- Remove all unnecessary code such as function calls to test your code
  → -1 point if it raises errors regardless of your functions or classes
- Remove or comment your *print().*
- After submitting your work, close your laptop. Then, you can see other books or paper-materials until the mid-term ends.

Information to solve the following problems:
[Tip] *str.upper()* returns a string where all characters are in upper case.
[Tip] *str.lower()* returns a string where all characters are in lower case.
[Tip] use help() if necessary (press 'q' button to quit from help() menial on the VS Code)
[Tip] press Ctrl+C (or Command+C on Mac) to escape from the infinite loop on VS Code.

Operator	Magic methods
+	__add__(self, other)
-	__sub__(self, other)
*	__mul__(self, other)
<	__lt__(self, other)
>	__gt__(self, other)
==	__eq__(self, other)

[Problem #1] (3 points)

- The function name: **function01**
- This function takes one argument
  - a *list* or *tuple* which has only int type elements
- The return value
  - Sum of elements which are odd. Return 0 if there is no element.
- For example, I will test the function using keyword argument, as follows:
  - function01(S=[1, 2, 3, 4, 5]) returns 9
  - function01(S=(1, 3, 2)) returns 4
  - function01(S=[]) returns 0
  - function01(S=[-2, 0, 1]) returns 1

[Problem #2] (3 points)

- The function name: **function02**
- This function takes one argument
    - str type (the length of this argument is greater than 1. All the characters are alphabet letters, no spaces)
- The return value (bool type)
    - *True* if the argument is palindrome, *False* otherwise. (ignore cases)
    - [Tip] The palindrome is a word that reads the same backwards as forwards, such as “우영우”, “기러기”, “인도인”, “level”, “mygym”, and “ABCcba”.
- For example, I will test the function using positional argument, as follows:
    - function02(“level”) returns True
    - function02(“levEL”) returns True
    - function02(“Aa”) returns True
    - function02(“hello”) returns False (because “hello” != “olleh”)
    - function02(“pythoN”) return False (because “pythoN” != “Nohtyp”)

[Problem #3] (3 points)

- The function name: **function03**
- This function takes the unknown number of positional arguments
    - all arguments are int type
- The return value
    - Sum of the arguments which are even. Return 0 if there is no argument.
- For example, I will test the function using positional argument, as follows:
    - function03() returns 0
    - function03(1) returns 0
    - function03(1, 2, 3) returns 2
    - function03(1, -2, 3, 6) returns 4

[Problem #4] (3 points)

- The function name: **function04**
- This function takes two arguments
    - first argument: int type
    - second argument: int type or str type
- The return value (float or Nonetype)
    - Return the divided value (divide the first argument by the second argument)
    - If any error or exception is raised, return None
- For example, I will test the function using positional argument, as follows:
    - function04(1, “10”) returns None
    - function04(1, 10) returns 0.1
    - function04(0, 0) returns None
    - function04(99, 99) returns 1.0

[Problem #5] (4 points)

- Please define **Person** class so that you can see the [Result] when running the [Code].
- **Person** class has two member variables
    - name: str type
    - height: int type

[Code] → I will test your class using this code using various name and height values

```
p = Person()
print("p.name:", p.name)
print("p.height:", p.height)
p.change_name("Kim")
p.change_height(170)
print("p.name:", p.name)
print("p.height:", p.height)
```

[Result]

```
p.name: Park
p.height: 160
p.name: Kim
p.height: 170
```

[Problem #6] (4 points)

- Please define **Animal** and **Dog** class so that you can see the [Result] when running the [Code].

[Code] → I will test your class using this code

```
a = Animal()
d = Dog()
if issubclass(Dog, Animal):
    print(a.cry())
    print(d.cry())
```

[Result]

```
Cry
Woof
```

[Problem #7] (5 points)

- Please define **Human** class so that you can see the [Result] when running the [Code]
- **Human** class has two member variables
    - name: str type
    - age: int type

[Code] → I will test your class using this code

```
h1 = Human()
h2 = Human("Kim", 30)
print("h1.name:", h1.name)
print("h1.age:", h1.age)
print("h2.name:", h2.name)
print("h2.age:", h2.age)
```

[Result]

```
h1.name: Park
h1.age: 20
h2.name: Kim
h2.age: 30
```

[Problem #8] (5 points)

- Please define **Number** class so that you can see the [Result] when running the [Code].
- **Number** class has one member variable
    - number: int type or float type

[Code] → I will test your class using the following code with various int or float values.

```
n1 = Number(1)
n2 = Number(3.15)
print(n1 + n2)  # n1 + n2 returns int or float type value
print(n1 - n2)  # n1 - n2 returns int or float type value
print("n1.number:", n1.number)
print("n2.number:", n2.number)
```

[Result]

```
4.15
-2.15
n1.number: 1
n2.number: 3.15
```

[Problem #9] (5 points)

- Please define **Student** class so that you can see the [Result] when running the [Code].
- **Student** class has one member variable
  - student_number: int type
- == compares student_numbers of the instances and returns True or False (bool type)

[Code] → I will test your class using the following code with various int values.

```
s1 = Student(1234)
s2 = Student(5678)
s3 = Student(1234)
print(s1==s1)  # s1==s1 returns bool type value, True or False
print(s1==s2)  # s1==s2 returns bool type value, True or False
print(s1==s3)  # s1==s2 returns bool type value, True or False
print(s1.student_number)
print(s2.student_number)
print(s3.student_number)
```

[Result]

```
True
False
True
1234
5678
1234
```

[Problem #10] (5 points)

- Please define **Account** class so that you can see the [Result] when running the [Code].
- **Account** class has one member variable
  - balance: int type

[Code] → I will test your class using the following code with various int values.

```
a1 = Account(0)  # initial money
a2 = Account(100)  # initial money
print("a1's balance:", a1.balance)
print("a2's balance:", a2.balance)
a1.deposit(20)  # deposit money into a1's account
a2.withdraw(10)  # withdraw money from a2's account
print("a1's balance:", a1.balance)
print("a2's balance:", a2.balance)
```

[Result]

```
a1's balance: 0
a2's balance: 100
a1's balance: 20
a2's balance: 90
```

- THE END -