연습문제 : 모듈, 패키지, 파일입출력, 예외처리

1. 다음 코드를 람다 함수 형태로 수정할 때, 알맞은 코드를 작성하시오. f = lambda x, y : x ** y

```
def f(x, y):
    return x ** y
```

2. 다음과 같이 리스트 컴프리헨션으로 되어 있는 코드를 람다(lambda) 함수와 map() 함수를 사용하여 표현하시오.

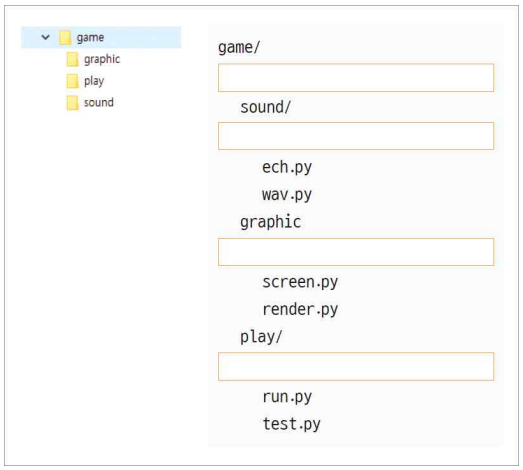
```
>>> f = lambda x : x ** 2
>>> list(map(f, ex))
```

```
>>> ex = [1, 2, 3, 4, 5]
>>> [value **2 for value in ex]
[1, 4, 9, 16, 25]
```

3. 패키지(packages)에 대한 설명 중 틀린 것은? ③

① 다양한 모듈의 합으로 디렉터리로 연결된다.
② 하나의 대형 프로젝트를 만드는 코드의 묶음이다.
③ 개별 .py 파일을 의미한다.
④ 다양한 오픈소스들이 관리되는 방법이다.
⑤ _ _init_ _, _ _main_ _ 등 키워드 파일명이 사용된다.

4. 'game'이라는 패키지를 만들고 싶다고 가정하자. 패키지를 만들기 위해 디렉터리별로 필요한 모듈을 구현하고자 한다. 다음 그림에서 빈칸에 들어가야 할 파일은?



① __main__ ② import game
③ __init__.py ④ __main__.py
⑤ __init__

5. 모듈을 호출하는 방법이 아닌 것은? ②

① import os ② import os as *
③ from os import listdir ④ from os import *
⑤ import os as linuxos

6. 두 코드 파일인 'fah_converter.py'와 'module_ex.py'는 같은 디렉터리에 있다. 다음과 같은 결과값을 얻기 위해 빈칸에 들어갈 적합한 코드를 쓰시오.

fah_converter.py

```
def covert_c_to_f(celsius_value):
    return celsius_value * 9.0 / 5 + 32

test_value = 0
```

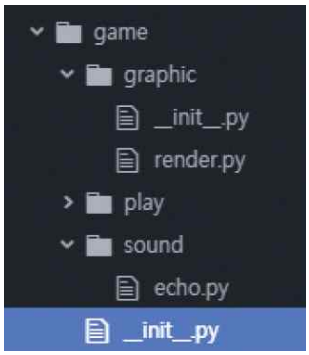
module_ex.py

```
import fah_converter as fah
print ("Enter a celsius value: ")
celsius = float(input())
fahrenheit = fah.covert_c_to_f(celsius)
print ("That's ", fahrenheit, " degrees Fahrenheit")
```

실행 결과

```
Enter a celsius value:
100
That's 212.0 degrees Fahrenheit
```

7. 다음과 같이 game 패키지를 만들었다. 모듈 render.py에 있는 render_test() 함수를 패키지 내에서 다른 디렉터리의 사용하려고 한다. 부모 디렉터리(game)를 기준으로 호출하는 방법은?



from game.graphic.render import render_test( )

8. 다음과 같이 'hello_python.txt' 파일을 수정하고자 한다. 빈칸에 들어가야 하는 코드를 쓰시오.

(가) a (나) write

수정 전 'hello_python.txt'

```
hello, world!
```

→ 수정 후 'hello_python.txt'

```
hello, world!
hello, python!
```

코드

```
f = open("hello_python.txt", '  (가)  ')
f.  (나)  ('hello, python!')
```

9. 파이썬은 파일 처리를 위해 open() 함수를 사용한다. "파일 열기 모드"에 따라 들어갈 값을 쓰시오.

```
f = open("<파일 이름>", "파일 열기 모드")
f.close()
```

(가) 쓰기 모드: w

(나) 추가 모드: a

(다) 읽기 모드: r

10. 'calculator_input.py'는 사칙연산 프로그램이다. 다음 빈칸을 채워 프로그램을 완성하시오.

calculator.py

```
def sum_func(a, b):
    return a + b

def multiply_func(a,b):
    return a * b

def minus_func(a,b):
    return a -b

def devide_func(a,b):
    return a / b
```

calculate_input.py

```
from calculator import *
user_input = input("사칙연산 프로그램: ").split(" ")
first_val, second_val=int(user_input [0]), int(user_input [2])
fourcal = user_input[1]

if fourcal == "+":
    result = sum_func(first_val , second_val)
elif fourcal == "-":
    result = minus_func(first_val , second_val)
elif fourcal == "/":
    result =devide_func(first_val , second_val)
else:
    result =multiply_func(first_val , second_val)
print("실행 결과는", result)
```

실행 예시

```
사칙연산 프로그램 : 5 * 4
실행 결과는 20
```

11. 다음 코드의 실행 결과를 쓰시오.

```
try:
    for i in range(1, 7):
        result = 7 // i
        print(result)
except ZeroDivisionError:
    print("Not divided by 0")
finally:
    print("종료되었습니다.")
```

```
7
3
2
1
1
1
종료되었습니다.
```

12. 다음 코드를 실행했을 때, 가장 마지막에 출력되는 값은?

```
sentence = list("Hello Gachon")
while (len(sentence) + 1):
    try:
        print(sentence.pop(0))
    except Exception as e:
        print(e)
        break
```

① o ② n ③ h

④ c ⑤ pop from empty list

13. 다음 각각의 예외 처리에 적합한 내장 예외(built-in exception)를 순서대로 실행한 결과값이 바르게 짝지어진 것은? ④

(가)	alist = ["a", "1", "c"] blist = ["b", "2", "d"] for a, b in enumerate(zip(alist, blist)): print(b[a])
(나)	alist = ["a", "1", "c"] blist = ["b", "2", "d"] for a, b in enumerate(zip(alist, blist)): print(a/int(b[0]))

① NameError, ValueError

② IndexError, NameError

③ ZeroDivisionError, ValueError

④ IndexError, ValueError

⑤ NameError, IndexError

14. 다음 중 예외(exception)의 이름과 내용이 잘못 짝지어진 것은? ④

① ZeroDivisionError: 0으로 숫자를 나눌 때

② ValueError: 변환할 수 없는 문자/숫자를 변환할 때

③ IndexError: 리스트의 인덱스 범위를 넘어 갈 때

④ SyntaxError: 조건문이나 변수에 오탈자가 존재할 때

⑤ NameError: 존재하지 않은 변수를 호출할 때

15. 파일의 종류에 대한 설명으로 틀린 것은? ④

① 바이너리 파일은 컴퓨터만 이해할 수 있는 형태인 이진법 형식으로 저장된 파일을 말한다.

② 텍스트 파일의 예로 HTML, 파이썬 코드 파일 등을 들 수 있다.

③ 바이너리 파일은 해당 확장자에 대한 파일을 열 수 있는 프로그램이 필요하다(엑셀, 워드 등).

④ 텍스트 파일의 경우 컴퓨터는 텍스트 파일 형태 그대로 처리가 가능하다.

⑤ 텍스트 파일은 사람도 이해할 수 있는 형태인 문자열 형식으로 저장된 파일을 말한다.

16. 다음과 같이 코드를 작성하고 실행하면 '숫자를 넣어주세요:'가 출력된다. 여기에서 문자를 입력하는 경우, 예측되는 실행 결과를 쓰시오. 숫자가 아닙니다.

```
import random
answer = random.randint(1,10)
def guess_number(answer):
    try:
        guess = int(input("숫자를 넣어 주세요: "))
        if answer == guess:
            print("정답!")
        else:
            print("틀렸습니다.")
    except ValueError:
        print("숫자가 아닙니다.")

guess_number(answer)
```

17. try ~ except 문에서 사용되는 예외의 종류에 대한 설명이다. 예외를 적어보자.

① 없는 변수에 접근할 때 NameError

② 파일 처리에서 오류가 발생할 때 IOError

③ 실행시에 오류가 발생할 때 RuntimeError

④ 딕셔너리에 키가 없을 때 KeyError

18. 다음 코드의 실행 결과를 쓰시오.

```
for i in range(3):
    try:
        print(i, 3// i)
    except ZeroDivisionError:
        print("Not divided by 0")
```

Not divided by 0

1 3

2 1

19. 다음 코드는 파이썬에서 'i_have_a_dream.txt' 파일을 읽어오는 코드이다. 같은 기능을 하는 코드를 with 구문과 함께 사용하여 작성하시오.

```
f = open("i_have_a_dream.txt", "r")
contents = f.read()
print(contents)
f.close()
```

```
with open("i_have_a_dream.txt","r") as my_file:
    contents = my_file.read()
    print(contents)
```

20. 'quiz.py'와 같이 코드를 작성한 후 저장하고, 파이썬 셸 코드를 실행했을 때의 결과값을 쓰시오.

quiz.py

```
def quiz():
    user_input = input()
    if int(user_input) > 3:
        print(user_input)
    else:
        print("user_input")
```

파이썬 셸

```
>>> import quiz as q
>>> q.quiz()
2
```

21. 다음 코드의 실행 결과를 쓰시오.

```
def sum_data(list_data_a, list_data_b):
    for i in list_data_a:
        for j in list_data_b:
            result = i+j
    return result

a = [1, 2]
b = [3, 4]
print(sum_data(a, b))
```

22. 파일과 폴더에 대한 설명으로 틀린 것은?

① 파일은 실행, 쓰기, 읽기 등을 할 수 있다.
② 파일은 컴퓨터에서 정보를 저장하는 논리적인 단위이다.
③ 10개 이상의 파일을 가진 폴더를 디렉터리라고 한다.
④ 폴더는 파일과 다른 폴더를 포함할 수 있다.
⑤ 파일은 파일명과 확장자로 식별된다.

23. 바이너리 파일과 텍스트 파일에 대한 설명으로 틀린 것은?

① 텍스트 파일은 사람도 이해할 수 있는 형태로 저장된다.
② 메모장에 저장된 파일, HTML 파일, 파이썬 코드 파일 등은 모두 텍스트 파일이다.
③ 텍스트 파일은 컴퓨터만 이해할 수 있는 형태인 이진(법) 형식으로 저장된 파일이다.
④ 엑셀 파일, 워드 파일 등을 바이너리 파일이라고 부른다.
⑤ 모든 텍스트 파일도 실제는 바이너리 파일로 아스키/유니코드 문자열 집합으로 저장된다.