

## 2024 BMED223 :: Hands-on instructions

- **Filename:** week10\_studentID\_studentName.zip
- **Deadline:** 2024/05/13 23:59
- **No excuse for late submission.** Prepare for submission in advance.
- There is no limit on the number of submissions, but grading will be based on the final file.
- Use comments (#, #%%, etc.) to separate problems.
- **Please write the classes in *prob1.py* that is separated from *main.py* file.**
- **And import the *prob1.py* into the *main.py* file and output the answer to the problem.**
- Please **compress all written python files** to the above name and submit.

서동휘 andy1200@korea.ac.kr

박영주 lime2514@korea.ac.kr

1. *programming\_class.txt*는 가공되지 않은 data를 저장하고 있다. 이를 file I/O를 통해 가공한 후 새로운 파일로 저장하라.

- 1) *programming\_class.txt*에 저장된 data는 아래와 같다.

### Contents of *programming\_class.txt*

```
,student no.,name,midterm,final,attendance,homework,total sum,grade,,  
,87530101,Kim TM,67,70,20,17.5,78.6,,  
,87530102,Ham HC,62,70,20,18.1,77.7,,  
,87530105,Cho HR,47,43,18,17.6,62.6,,  
,87530107,Lee MY,25,18,20,18.4,51.3,,  
,87530108,Kim TY,65,36,20,17.6,67.9,,  
... (continued)
```

- 첫 번째 line은 data의 column 정보이다.
- 두 번째 line부터 student 별 data이다. 이 때, comma(',')로 column 값이 구분되며, 마지막 항목 grade의 값은 비어있다.

- 2) Goal : 아래의 규칙에 따라 program을 작성하여 grade를 산출하고 새로운 file에 기록하라.

- i. Score class를 정의하여 모든 학생의 점수를 관리해야 함.
  - 학번, 이름, 점수, 출석 등을 Score class의 attribute로 설정.
  - 각 점수를 dictionary와 list를 이용하여 구조를 만들어야함.
  - Student class에 grade를 산출하는 method를 정의한다. Grade를 산출하는 방식은 자유롭게 정하여 구현한다.
  - 그 밖에 필요하다고 생각하는 attributes와 methods, 더 나아가 필요하다고 생각하는 새로운 class를 추가로 구현해도 된다.

- ii. *main.py* 에서 위에서 만든 class를 읽어, class instance를 생성하고 이를 통해 새로운 file에 score 정보를 저장한다.
  - 각 학생 row의 끝에 grade가 기재된 최종결과 파일을 *programming.grade.txt* 로 저장한다.
  - Column에 따른 값의 위치가 올바르게 규격을 맞추도록 한다. (완벽하게 맞출 필요는 없음, hint = string formatting)

### 3) Result

Contents of <i>programming_grade.txt</i>							
student no.	name	midterm	final	attendance	homework	total sum	grade
87530101	Kim TM	67	70	20	17.5	78.6	B
87530102	Ham HC	62	70	20	18.1	77.7	B
87530105	Cho HR	47	43	18	17.6	62.6	C
87530107	Lee MY	25	18	20	18.4	51.3	D
87530108	Kim TY	65	36	20	17.6	67.9	C
... (continued)							

## 2. 실험을 통해 얻은 data를 효과적으로 정리하기 위해 json format을 이용하려고 한다.

- 1) Data의 모습은 아래와 같다.

Contents of <i>prob2.dat</i>					
241	0.7	0.66	-9.84	-2.42	
241	0.7	0.66	-5.85	7.41	
133	0.7	0.66	11.05	-3.4	
457	0.7	0.66	3.53	-5.18	
457	0.7	0.66	2.36	-3.82	
457	0.7	0.65	-1.15	-1.37	
457	0.7	0.66	-4.07	11.22	
... (continued)					

- Line 마다 하나의 data 의 정보를 나타내고 있으며, data 의 정보는 차례대로 idNumber, time, energy, x-coordinate, y-coordinate 를 의미한다.

- 2) 위의 data 정보의 각 column을 이용하여 다음과 같이 json formatting하여 저장하여라.

Json format
<pre>{ "133": [     {         "time": 0.7,         "energy": 0.66,         "coordinate": {             "x": 11.05,             "y": -3.4         }     },     ... (more) ] '241': [     ... (이하 생략) ] '457': [     ... (이하 생략) ] }</pre>

- idNumber를 key로, idNumber에 해당하는 data를 담은 list를 value로 하여 저장한다.
- List에 담은 data의 정보는 위와 같이 dictionary 형태로 각각 저장된다.
- Json module을 이용하여 새로운 파일(*prob2.json*)로 저장한다.

3) Load json data

- 저장된 *prob2.json*을 불러와 idNumber가 '133'인 data 5개를 print해보아라.

result
<pre>[{'time': 0.7, 'energy': 0.66, 'coordinate': {'x': -8.22, 'y': 2.56}}, {'time': 0.7, 'energy': 0.66, 'coordinate': {'x': 6.84, 'y': -3.88}}, {'time': 0.7, 'energy': 0.66, 'coordinate': {'x': 9.66, 'y': -6.4}}, {'time': 0.7, 'energy': 0.66, 'coordinate': {'x': -4.25, 'y': -3.35}}, {'time': 0.7, 'energy': 0.66, 'coordinate': {'x': -3.47, 'y': -2.86}}]</pre>