

## Lab #5 - 학점 계산기 (exam\_grader)

Copyright 2015 © document created by TeamLab.Gachon@gmail.com

### Introduction

이번 과제는 조금 짧다. 그러나 간단하지 않다. 처음으로 수강자가 직접 함수를 작성한다. 나중에 보면 매우 쉬운 함수 이지만 지금은 어렵게 느껴질 것이다. 이미 포기할 순 없다. 해보자.

### 숙제 template 파일 다운로드

먼저 숙제 template 파일을 cs50 서버로 부터 다운로드 받는다. 로그인 후 나타나는 bash shell에서 다음과 같은 명령을 입력하자.

```
python3.4 submit_assignment.py -get exam_grader
```

exam\_grader.py 파일이 다운로드 될 것이다. 받자마자 실험 차원에서 코드를 실행해보자. python3.4 exam\_grader.py를 입력하면 된다. 이제 다들 코드를 어떻게 실행하는 지는 알 것이다. 실행하면 아마 아래와 같은 에러 메시지가 뜰 것이다.

### Start of Exam Grader Program

```
=====
```

Traceback (most recent call last):

```
File "exam_grader.py", line 67, in <module>
    main()
```

```
File "exam_grader.py", line 58, in main
```

```
    total_score = sum_of_scores(number_of_subjects)
```

```
File "exam_grader.py", line 29, in sum_of_scores
```

```
    for i in range(number_of_subjects):
```

TypeError: 'NoneType' object cannot be interpreted as an integer

기존의 lab과 달리 이번 코드는 각 함수들을 개별로 작동하는 코드가 아닌 함수 들이 순차적으로 작동하여 최종적인 목표를 달성하도록 설계되어 있다. 코드의 구조를 살펴보자.

### exam\_grader.py 코드 구조

이번 lab을 수행하기 위해서 첫 번째로 main 함수부터 들여다 보자. main 함수는 아래와 같이 구성되어 있다. 혹시몰라 또 적지만 밑에 코드는 vim 로 확인이 가능하고 vi exam\_grader.py 명령으로 파일을 열어볼 수 있다.

```
def main():
```

```
    print("Start of Exam Grader Program")
```

```
    print("=====")
```

```
    number_of_subjects = get_number_of_subjects()
```

```
    total_score = sum_of_scores(number_of_subjects)
```

```
    average_score = get_average_score(
```

```
        total_score=total_score, number_of_subjects=number_of_subjects)
```

```

print_exam_grader(average_score)

print("=====")
print("End of Exam Grader Program")

```

처음 두줄과 마지막 두줄은 프로그램의 시작과 끝을 알리는 print문으로 실제 프로그램 실행에 영향을 주지 않는다. `number_of_subjects = get_number_of_subjects()` 코드에는 우리가 첫 번째로 수정해야 할 함수가 나온다. 본 코드는 `get_number_of_subjects`의 함수에서 나온 결과 값을 `number_of_subjects` 변수에 저장하라는 의미이다. `get_number_of_subjects`는 사용자에게 입력을 받아 `total_score`를 계산해준다. 다음 줄인 `total_score = sum_of_scores(number_of_subjects)`는 실제 이번 숙제에 직접적인 영향을 주지 않는 helper 이다. 전체적인 프로그램의 실행을 도와주는 함수로 과목의 수에 따라 각 과목의 점수를 입력받을 수 있도록 설계되어 있다. 숙제 제출이 끝난 후 살펴볼 것을 권한다. 다음 줄인 `average_score = get_average_score(total_score=total_score, number_of_subjects=number_of_subjects)` 코드가 이번 lab에서 가장 중요한 코드이다. 본 코드는 `get_average_score`라는 함수에 `total_score`와 `number_of_subjects`라는 변수를 입력하여 성적의 평균 값을 `average_score`에 할당한다. `number_of_subjects`는 첫 번째 코드에서, `total_score`는 두 번째 코드에서 각각 값이 결정된다. 현재 다운로드 받은 lab code에는 `get_average_score`가 존재하지 않는다. 직접 작성해야만 한다. 마지막으로 `print_exam_grader(average_score)`는 성적의 평균 값을 바탕으로 최종 평균과 학점을 화면에 출력해주는 함수를 사용하는 코드이다. 역시 helper 로 수강자가 수정할 필요가 없다. 그러나 다음 시간에 배울 if문에 대해서 적혀있으므로 숙제 제출후 꼭 살펴볼 것을 권한다.

## get\_number\_of\_subjects 함수 수정하기

첫 번째 수정내용은 `get_number_of_subjects` 함수이다. 본 함수의 코드는 아래와 같다.

```

# Console
def get_number_of_subjects():

    # """
    # Input:
    # - None
    # Output:
    # - number_of_subjects: Integer Type
    # Examples(python shell):
    # >>> import exam_grader as eg
    # >>> eg.get_number_of_subjects()
    #      : 10
    # 10
    # """
    #
    # ==Modify codes below=====
    number_of_subjects = None
    # =====
    return number_of_subjects

```

복잡해 보일 수도 있으나, 함수 자체는 간단하다. 입력 받는 값은 없고 함수를 실행 시키면 사용자 console 창에 `10` : 라는 질문이 나오면서 사용자의 입력을 기다린다. 사용자가 정수 입력하면 그 값을 integer type으로 변환하여 반환해준다. 간단하게 1) 입력 받고, 2) 정수로 변환해주기만 하면 되는 함수이기 때문에 쉽게 할 수 있을 것이다.

## get\_average\_score 함수 생성하기

이번엔 전혀 주어진 template 코드 없이 직접 함수를 작성해보자. 함수에 대한 정보는 아래와 같다.

내용 | 구성 --- | --- 함수명 | get\_average\_score input 변수 | 1. total\_score : Integer Type의 성적 총합 | 2. number\_of\_subjects : Integer Type의 과목 갯수 output 값 | Float Type의 total\_score을 number\_of\_subjects로 나눈 값

이것만 보고 어떻게 함수를 만들지 대략 난감하겠지만, 이정도 정보면 모든 정보를 다 준 것이다. 실제 개발을 할 때는 함수명이나 변수명은 개발자가 임의로 정할 수 있다. 그러나 본 숙제에서는 정해진 규칙대로 작성하지 않으면 숙제 검사를 해주지 않는다. 컴퓨터는 거짓말을 하진 않지만, 그렇다고 융통성이 있지도 않다.

## 테스트 및 제출

두 함수를 모두 작성했다면 실행을 해보자. bash shell에서 python3.4 exam\_grader.py 명령어로 실행을 하면된다. 정확히 작성되었다면 아래와 같이 결과를 볼 수 있을 것이다. 아래 결과중 셋째 줄인

:는 출력되고 뒤에 3은 실습자가 직접 입력하는 것이며, 그 아랫줄에 85,95,100 모두 사용자가 직접 입력하는 항목이다.

Start of Exam Grader Program

```
=====
: 3
1      : 85
2      : 95
3      : 100
: 93.33333333333333
: A
=====
```

End of Exam Grader Program

이제 숙제를 python3.4 submit\_assignment.py -submit exam\_grader.py 명령어를 입력하여 제출하자. 완벽하지 못한 상태에서 숙제 제출을 수행하면 아래와 같은 에러를 볼 수 도 있을 것이다.

Function Name	Passed?	Feedback
get_number_of_subjects	PASS	Good Job
get_average_score	Not Yet	Check Your Grammar

이럴 경우에는 get\_average\_score 함수를 이름에 안맞게 작성했을 경우 주로 나타난다. 물론 아닌경우도 있다. 다시 위의 내용을 점검하여 오타자가 없는지 확인한 후 제출하자. 제대로 작성할 경우 아래 메시지를 볼 수 있다.

Function Name	Passed?	Feedback
get_number_of_subjects	PASS	Good Job
get_average_score	PASS	Good Job

## Next Work

느끼겠지만, 더 이상의 친절함은 없다. 점점 스스로 혼자 헤쳐나가야 한다. 많은 에러 메시지와 사투를 벌여야 하고, 웬지 모르지만 안되는 이유를 찾아야 한다. 이제 구글에 익숙해 지길 바란다. 구글 검색이야말로 최고의 프로그래머가 되는 지름길이다. 마지막 lab 으로 가자!

Human knowledge belongs to the world – from movie ‘Password’ –

## Footnotes