CS2204 Homework: Group Assignments

Objectives

- Learn to read existing Python code
- Find and correct *syntax errors* in code
- Find and correct *logical errors* in code
- Understand PEP 8 style requirements

Background

Your instructor needed to assign his students to three synchronous classes, so each student will come to class either on *Monday*, *Wednesday* or *Friday*. Traditionally, professors use a simple lexicographical ordering scheme, i.e. all students with last names between "A" to "H" are assigned to the Monday class, and so on.

However, this created biased classrooms, assigning too many international students to the same day. Thus, the instructor created a Python program which reads the list of students from an external file (`classroll.txt`), shuffles the list and assigns the shuffled list of students to the groups. It is important that

- **the assignments are randomized**
- **each student is assigned to exactly one day**
- the days are balanced: **the maximum difference between the number of students in any two groups is 1**

Also, the results of the program is supposed to look like this

```
Monday
    Student One
    Student Two
    ...

Wednesday
    Student Three
    Student Four
    ...

Friday
    Student Five
    Student Six
    ...
```

Unfortunatelly, your instructor is a terrible Python programmer, and made several mistakes along the way. Your task is the fix these errors.

Tasks

1. There are a few syntax errors in the provided script (`group_assignment.py`). You have to fix this first (30 pts)
2. There is a logical error in the program. It does not implement the assignment properly (see the requirements, above). Understand and fix this problem by modifying the existing code, or - if it feels easier - by completely rewriting the algorithm (50 pts)
3. Make sure, you maintain the proper output format. The current script might do this right, just make sure you do not ruin it. (20 pts)
4. There are several PEP 8 style errors in the code. Fix these, otherwise you will be penalized.

Hints

There are a few constructs in the code, which might be new to you. While most of these are not critical to understand to fix the program, here are some brief explanations (we learn about these more, later).

`random.sample(population, k)` : it picks `k` number of samples (without replacement) from `population` . Both `population` and the returned samples are Python lists. If `k` equals to the length of the `population` list, the result is effectively a randomly shuffled version of that list.

Example:

```
1 from random import sample
2
3 vals = [1, 2, 3, 4, 5, 6, 7]
4 shuffled = sample(vals, len(vals))
5 print(shuffled)
```

`zip(list_a, list_b, ...)` : returns a new list (more properly, an iterator object), containing tuples by merging the original lists elementwise. The most typical use-case is with `for` loops, so we can iterate over the elements of multiple lists in parallel.

Examples:

```
list_a = "one", "two", "three"
list_b = "uno", "dos", "tres"
print(list(zip(list_a, list_b)))
```

```
for a, b in zip(list_a, list_b):
    print(a, "--", b)
```

Grading

You can use the attached `validator.py` program to check your work (and the instructor's original mistakes). It will also estimate your final score for the homework.

Penalties

Points will be deducted if you fail to set `__author__` variable (-10 pts) and for **each PEP 8 style errors** (-1 pt for each) in your program.

Submission

Please, upload the final version of the following file(s) (**and only those file(s)**) to Brightspace:

- `group_assignment.py`