NATIONAL UNIVERSITY
OF COMPUTER & EMERGING SCIENCES
PESHAWAR CAMPUS

Problem Set:	Assignment: AG06	**Semester**:	Fall 2018
Points:	*See autograder*		
Date Set:	*See autograder*	**Due Date**:	*See autograder*
Course:	CS118 Prog. Fundamentals	**Instructor**:	Dr. Nauman

1 Lists, Tuples and Dictionaries

Since you are reading this, you have already downloaded and extracted the zip file.
Read through the whole description below before starting with the assignment.

1.1 Tasks to do

1. Take a look at `a06.py`

2. There are two main tasks to complete.

   (a) Write a function with the name `get_student_marks`. This function will be given the following list as its only parameter:

```
[
  {'roll_no': 'p18-1001', 'marks': {
      'english': (1.4, 2.5, 15, 9.6, 33),
      'calculus': (2.4, 1.5, 12, 1.6, 21),
    }, 'attendance': 88.4
  },
  {'roll_no': 'p18-1002', 'marks': {
      'english': (2.4, 1.5, 12, 1.6, 21),
      'programming fundaments': (2.4, 1.5, 12, 1.6, 21),
    }, 'attendance': 79.4
  },
  {'roll_no': 'p18-1003', 'marks': {
      'calculus': (2.4, 1.5, 12, None, 21),
      'programming fundamentals': (2.4, 1.5, 12, 1.6, 21),
    }, 'attendance': 79.4
  },
]
```

   This is not that complicated. So, pay attention: It's a list of student records – each record being a dictionary. This has three keys: roll_no and attendance are straight-forward but the value of the key 'marks' is itself a dictionary.

   This dictionary has subject names as keys and tuples as values. Each of these tuples has the marks the student has obtained in a quiz/assignment. (If the student did not take that quiz, the record will have a `None` there.)

   Your mission, should you choose to accept it, is to write `get_student_marks` in a way that it returns the cumulative marks of each student but in a specific structure: So, for the case above, it should return a dictionary as follows:

```
{
 'p18-1001': {'english': 61.5,
              'calculus': 38.5
             },
 'p18-1002': {'english': 38.5,
              'programming fundaments': 38.5
             },
 'p18-1003': {'calculus': 36.9,
              'programming fundamentals': 38.5
              }
}
```

So, you now have a dictionary where the key is the student's roll number and the value is a dictionary of subject, total marks pairs.

Your function should be able to handle any number of evaluation marks in the tuple as well as any number of students.

You can go ahead and write this function before reading the rest of the PDF.

(b) The second function you need to write is `get_grade`. It will be given a float as its first argument and a dictionary as the second argument. For instance, we can send it the marks 65 as its first argument and the dictionary sent to it will be, for example, the following:

```
{'A': 80, 'B': 70, 'C': 60, 'D': 50}
```

The semantics of this dictionary are that if the student gets more than or equal to 80 marks, the grade should be 'A'. With 70, the grade should be B and so on. Anything below a 50 is an F.

So, for the above two parameters, the grade should be C. Notice that the dictionary is only an example and the values of these grade can change (but the actual grade letters will always be A, B, C, D or F). You can assume that marks for A will always be greater than those for B and so on.

3. Run local tests and if they pass, submit the assignment using the submission command given on the Autograder assignment page. (Same as the first assignment.)