【2023 Object-Oriented Programming Homework 1】

1. Please use **Python language** for this homework and make sure it can run correctly.
2. Please provide **ipynb files** to validate your homework.
3. Do not copy the others work definitely. Otherwise, you will fail this class.
4. If you have any question, please send email to TA or drop by Room EC5018. However, TA will not help you to debug program.

Turn in your homework:

1. Please compress your homework into zip file.
2. Naming rule: "OOP_HW1.zip".
3. Upload your homework (zip file) to NSYSU Cyber University (網路大學).
4. **Deadline: 2023/03/03 09:00.** You can not get any credit if you do not turn in your homework before the deadline.

Homework description:

1. Write a program that predicts the score needed on a final exam to pass the course. The program should interact with the user as follows:

   Enter course name> Object-Oriented Programming

   Enter minimum average required> 79.5

   Enter current average in course> 74.6

   Enter how much the final counts

   as a percentage of the course grade> 25

   You need a score of 94.20 on the final to pass 'Object-Oriented Programming'.

2. Given a string consisting of words and spaces, return the length of the first and last word in the string. The program should interact with the user as follows:

   Enter string> Doge to the moon!

   The first word is 'Doge' with length 4.

   The last word is 'moon!' with length 5.

3. Write a program to add item in the following Python List. You have to reproduce the same list by yourself. They will look like the following lists.

   list = [10, 20, [300, 400, [5000, 6000], 500], 40, 30]

   Add 7000 after 6000: [10, 20, [300, 400, [5000, 6000, 7000], 500], 40, 30]

   Add 600 before 500: [10, 20, [300, 400, [5000, 6000, 7000], 600, 500], 40, 30]

4. Write a program to add sublist in a nested list. You have to reproduce the same nested list and sublist by yourself. They will look like the following lists.

   list = ['a', 'b', ['c', ['d', 'e', ['f', 'g'], 'k'], 'l'], 'm', 'n']

   sub_list = ['h', 'i', 'j']

   new list1 = ['a', 'b', ['c', ['d', 'e', ['f', 'g', 'h', 'i', 'j'], 'k'], 'l'], 'm', 'n']

   new list2 = ['a', 'b', ['c', ['d', 'e', ['f', 'g', ['h', 'i', 'j']], 'k'], 'l'], 'm', 'n']

5. Write a program to calculate BMI and print message of each person. There are three fields (name, height and weight) for user to input. You have to capitalize the first letter of each word in the name string and lowercase the rest. After completing the input, save information in nested list and print all data including previous input. The formula for BMI is weight in kilograms divided by height in meters squared. If BMI is less than 18.5, it falls within the underweight range. If BMI is

18.5 to <25, it falls within the healthy weight range. If BMI is 25.0 or higher, it falls within the overweight range. You have to display the BMI value with one decimal place and corresponding message. The program should interact with the user as follows (Hint: You can learn **for loop** and **if-else statement** from section3):

Enter number of people> 3

(1) Enter name> Vivian

(1) Enter Vivian's height(cm)> 155

(1) Enter Vivian's weight(kg)> 42

All data: [['Vivian', 155.0, 42.0]]

(2) Enter name> rex

(2) Enter Rex's height(cm)> 160

(2) Enter Rex's weight(kg)> 50

All data: [['Vivian', 155.0, 42.0], ['Rex', 160.0, 50.0]]

(3) Enter name> Henry smith

(3) Enter Henry Smith's height(cm)> 170

(3) Enter Henry Smith's weight(kg)> 85

All data: [['Vivian', 155.0, 42.0], ['Rex', 160.0, 50.0], ['Henry Smith', 170.0, 85.0]]

==========

Vivian's BMI is 17.5

Vivian is underweight.

Rex's BMI is 19.5

Rex maintains optimal body weight.

Henry Smith's BMI is 29.4

Henry Smith is overweight.