

## HOMEWORK 2

### COP3035 – Intro Programming in Python - Summer 2024

Due date: 6/7/2024

#### Instructions:

- Submit your answers to questions in a single consolidated PDF file to Canvas.
- Show code and outputs that demonstrate your solutions.
- Multiple submissions are allowed; however, only the final submission made before the due date will be graded.
- No late submissions will be accepted, make sure you plan to submit at least 2hrs before the deadline to accommodate for unexpected technical difficulties.
- Once you've finished the Jupyter Notebook tasks, export the notebook as a PDF using the export feature (don't send the original .ipynb file or change the extension).
- Make sure you submit the right file to Canvas. Wrong file submissions will not be graded.
- Note: Inquiries about homework must be sent to the instructors within 3 days after grades are published.

#### Objective:

You will create a Python program that computes the final letter grade based on individual grades from different categories (labs, homework, exams, and participation). Your task is to use the provided weights and grade conversion table to determine the letter grade. The program should also print a detailed grade report.

1. **Create four lists:** labs, homework, exams, and a variable for participation grade. Each list will contain numerical grades as input.

Example:

```
labs = [100, 50, 0, 100, 50, 100, 100, 100, 50, 0, 1]
homework = [110, 90, 80, 110, 70, 0, 120]
exams = [90, 100, 80, 100]
participation_grade = 100
```

2. **Compute Category Grades:**

Calculate the average for each category (labs, homework, and exams).

3. **Compute Total Grade:**

Use the given weights to calculate the total grade.

4. **Determine Letter Grade:**

Use the grade conversion table to find the corresponding letter grade for the computed total grade.

Tip:

```
# Find the letter grade
l = list(grade_table.keys()) # List of intervals tuples
filter = [l[0][0] <= grade <= l[0][1],
          l[1][0] <= grade <= l[1][1],
```

```

        l[2][0] <= grade <= l[2][1],
        l[3][0] <= grade <= l[3][1],
... etc....
        l[9][0] <= grade <= l[9][1] ]

letter_grade = grade_table[l[filter.index(True)]]

```

### 5. Print the Grade Report:

Display a detailed report showing all individual grades, category averages, weighted contributions, total grade, and letter grade.

#### GRADE REPORT

```

-----
Homework grades: [110, 90, 80, 110, 70, 0, 120] = 580/7 = 82.9
Lab grades: [100, 50, 0, 100, 50, 100, 100, 100, 50, 0, 1] = 651/11 = 59.2
Participation grade: 100.0
Exam grades: [90, 100, 80, 100] = 370/4 = 92.5
-----
Homeworks      : 82.9 x 0.40
Labs           : 59.2 x 0.05
Participation   : 100.0 x 0.05
Exams          : 92.5 x 0.50
-----
TOTAL          : 87.4
-----
GRADE          : B+
-----

```

6. **Bonus (10 points).** If the code automatically computes the letter grade and the detailed report by changing any of the values on step 1.

#### Requirements:

**Do not use loops, functions, conditionals, or object-oriented programming (OOP) constructs. If you use them we will discount points.**

Use only the concepts covered in class: strings, lists, dictionaries, and tuples.

You may use functions like `sum()`, `len()`, `list()`, and dictionary operations.

Given Data:

a) Weights dictionary:

```
weights = {'Labs': 5/100, 'Participation': 5/100, 'Homeworks': 40/100,
           'Exams': 50/100}
```

b) Letter grade table dictionary:

```
grade_table = {
    (93, 100): 'A',
```

```
(90, 92): 'A-',  
(85, 89): 'B+',  
(80, 84): 'B',  
(75, 79): 'B-',  
(72, 74): 'C+',  
(68, 71): 'C',  
(60, 67): 'C-',  
(50, 59): 'D',  
(0, 49): 'F'  
}
```