Koç University COMP125
Programming with Python: Midterm 1
Instructor: Beren Semiz
Date: March 21st, 2021

12:10 PM – 1:40 PM
Deadline to submit on Blackboard: 1:45 PM

Instructions

Make sure you read and understand every part of this document

Download & Extract the zip to your Desktop:

Blackboard -> Assessments -> MT1-> MT1.zip

Zip file includes:

multipleChoice.py triangleTypesMT.py calculateSeries.py Midterm1_Spring2021.pdf

Open the .py files using Spyder.

Do not change the names of the files. Do not change the names of the functions that are given to you. Do not use Turkish characters. You may add new functions if you want to (helper functions).

When you are finished:

Submit a single compressed archive (.zip or .rar file) containing:

multipleChoice.py triangleTypesMT.py calculateSeries.py

Multiple attempts are allowed, so upload ASAP, clean up later. We will grade only the LAST version submitted to Blackboard.

By submitting this file, you automatically agree to the honor pledge: "You certify that you have completed this exam on your own without any help from anyone else. You understand that the only sources of authorized information in this exam are (i) the course textbook, (ii) the material that is posted at Blackboard for this class, and (iii) any study notes handwritten by yourself. You have not used, accessed or received any information from any other unauthorized source in taking this exam. The effort in the exam belongs completely to you."

Communication before & during exam:

- Zoom session (Meeting ID: 997 3064 5604 Passcode: comp125)
  - Starts at 12:00pm
  - Students are not allowed to speak or chat via this channel once the exam starts.
  - Violations may be punished via disciplinary action !!!
- For clarification questions:
  - Raise your hand in Zoom.
  - We will transfer you to a breakout room.
  - Only clarification questions regarding the midterm are allowed.
  - Installation related or my_code_is_not_working questions will not be answered.

Q1: Multiple Choice - 30 pts

In your science class, your instructor decided to prepare a multiple-choice exam including exactly 25 questions. Each question is worth 4 points and maximum grade is 100. Your answer to each question could be correct, wrong or blank. However, since your instructor wants to prevent random guessing, she is thinking of two different solutions:

1. In the first solution, **for each wrong answer you will lose 0.25 of a correct answer**. Few examples are given in the table below.

Correct	Blank	Wrong	Penalty	Final grade
24	0	1	0.25	95
24	1	0	0	96
23	0	2	0.5	90
19	1	5	1.25	71
17	2	6	1.5	62

Implement a function called decimal_deduction(correct, wrong) which takes the number of correct and wrong answers as inputs and calculates the final grade for the student. Your function should **return** the final grade. **(15 pts.)**

Few test cases for you:
decimal_deduction(24, 1) should return 95
decimal_deduction(23, 2) should return 90
decimal_deduction(19, 5) should return 71
decimal_deduction(17, 6) should return 62

2. In the second solution, **for four wrong answers, you will lose one correct answer**. In this rule, there won't be any decimal deductions. Few examples are given in the table below:

Correct	Blank	Wrong	Penalty	Final grade
24	0	1	0	96
24	1	0	0	96
23	0	2	0	92
19	1	5	1	72
17	2	6	1	64

Implement a function called whole_deduction(correct, wrong) which takes the number of correct and wrong answers as inputs and calculates the final grade for the student. Your function should **return** the final grade. **(15 pts.)**

Few test cases for you:
whole_deduction(24, 1) should return 96
whole_deduction(23, 2) should return 92
whole_deduction(19, 5) should return 72
whole_deduction(17, 6) should return 64

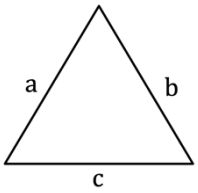
RULES:

- You can assume that in both functions, the total of correct and wrong answers will not exceed 25. You do not need to check for erroneous inputs.
- The student's final grade cannot be below 0. If the student's grade falls below 0, your functions should return 0.

Q2: Triangle Types MT - 35 pts

In geometry, all three of the following conditions should hold so that given sides can form a triangle:

a + b > c
a + c > b
b + c > a



Your task is to write a function to check triangle types based on the side lengths.

You should first check whether the given side lengths form a triangle or not.

- If the given sides do not form a triangle, **return** -1

If they do form a triangle, then you need to determine the type of the triangle based on the following rules:

- If the triangle has three equal sides (equilateral triangle), **return** 1
- If the triangle has two equal sides (isosceles triangle), **return** 2
- If the triangle has no equal sides (scalene triangle), **return** 3

Open triangleTypesMT.py and implement your code inside triangle_typeMT(a,b,c) function. Return values should be integers. Do not return them in string form.

Some test cases for you:
triangle_typeMT(5,7,6) should return 3
triangle_typeMT(6,3,2) should return -1
triangle_typeMT(4,4,4) should return 1
triangle_typeMT(8,8,6) should return 2

Q3: Calculate Series - 35 pts

Write a function compute_series(x, terms) that computes and **returns** the value of the following series:

$$cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \frac{x^8}{8!} - \cdots$$

Your function has two inputs:

- x: this is the value for x that will be used in the calculations
- terms: denotes how many terms should be considered in the series. In the above example, terms=5. There may be fewer terms or more terms.

Open calculateSeries.py, implement your code under compute_series(x, terms) function. If you want, you can define a helper function such as compute_factorial(n).

HINTS:

- Your result will eventually converge to cos(x)
- Think about the pattern first!
    - What is the relationship between the index and the power?
    - What is the relationship between the index and the sign?

Some test cases for you:
compute_series(1,3) should return 0.5416666666666666
compute_series(1,5) should return 0.5403025793650793
compute_series(1,20) should return 0.5403023058681397
compute_series(1,100) should return 0.5403023058681397

Submission + Honor Pledge

Solve each question in its own file. **DO NOT CHANGE THE NAMES OF THE FILES. DO NOT CHANGE THE HEADERS OF THE GIVEN FUNCTIONS (FUNCTION NAMES, FUNCTION PARAMETERS). DO NOT USE TURKISH CHARACTERS.**

When you are finished, compress your MT1 folder containing all of your answers:
multipleChoice.py triangleTypesMT.py calculateSeries.py

The result should be a SINGLE compressed file (file extension: .zip or .rar). Upload this compressed file to Blackboard.