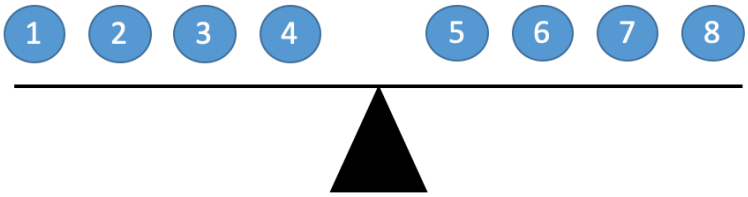
Odd Ball – 2022



OVERVIEW

In this assignment, you are going to design and develop a python program to run an interactive odd-ball game. The game is composed of an even number of balls and a single scaling device. All balls look identical in size and appearance, and all balls weigh exactly the same except one ball, which is heavier, called the odd ball. Balls are identified by their number labels in consecutive numerical sequence, starting with 1. The scale is a balance instrument with a left pan and a right pan used for weighing, the beam is in balance when the 2 pans contain exactly the same mass. The implemented program will interact with a player and the player will try to guess the odd ball by repeatedly weighing different balls on the scale. During the game the program and the player will take turns in an alternative manner: the player chooses some balls to weigh, the program outputs the result and then prompts the user to make a guess for the odd ball.

At the start of the program, it first prompts the player to enter the number of balls for the game. The number of balls must be even, minimum 2. After the prompt, the game randomly chooses one ball as the odd ball and keeps it secret from the player. Subsequently the game prompts the player for the balls to be placed on the left pan of the scale, and for the balls on the right. In return the game outputs the result of the weighing as whether the mass on both pans are balanced, left pan is heavier or right pan heavier. Finally, the player makes a guess for the odd ball. If the guess is not correct, the game will prompt the player to weigh again and the same process repeats until a correct answer is entered. When a correct guess is received, the game will output the total number of guesses made and total number of times the scale was used.

SCOPE

- At the start of the program, display a brief introduction about the game.
- For each game, prompt player for the number of balls, the number must be even, minimum 2.
- Implement the Game Cycle:
    a. Prompt user for ball weighing.
    b. Output the result of the scale
        i. Balanced
        ii. Left Heavier
        iii. Right Heavier
    c. Prompt user to make a guess for the odd number
- Track total number of times the scale used for each game.
- Display the scale usage count after the correct guess is entered.
- At the end of the game, prompt player to quit or start a new game
- Input Validation (User Interface)
    a. Ensure correct input is entered, if not, display prompt again after showing the warning.
    b. Robustness: program should not crash due to invalid input
- Coding Styles
    a. Ensure that your program follows the proper layout structure as discussed in class.
    b. You might declare global variables used for this assignment, ensure that a consistent naming convention is in place to differentiate various variable scopes.

NOTE:

- Keep your entire source code in ONE SINGLE file.
- Use only standard python modules
- In your design stick ONLY to functions, in other words, no class objects of your own.

STARTUP OPTIONS

Not applicable

SKILLS

In this assignment, you will be trained on the use of the followings:

- Understand requirements as per scope
- Use standard python module to design the game program as per scope
- Use standard objects (strings, numbers & lists)
- Variable Scope
- Functions for program structure and decomposition
- User Interface (Interaction)
- Input Validation

DELIVERABLES

1. Program source code (A1_School_StudentID.py)

where School is SSE, SME, HSS, FE or LHS and StudentID is your 9-digit student ID.

Submit the python file by due date to the corresponding assignment folder under “Assignment (submission)”

For instances, a SME student with student ID “119010001” will name the program as follows:

- A1_SME_119010001.py:

5% will be deducted if file is incorrectly named!!!

LATE SUBMISSION

For late submission, a daily penalty amounting to 10 points will be deducted against the final grade, up to maximum 3 days (30 points).

No further submission will be accepted 3 days after the set deadline.

TIPS & HINTS

- Follow the layout structure as mentioned in class (import, declarations, functions, main process).
- Clearly name and comment your global variables.
- Refer to python website for program styles and naming convention (PEP 8).

SAMPLE USER INTERFACE

```
        Welcome to Kinley's odd-ball game! You are given
        an even number of balls, labelled, and among
        the balls one is heavier than the rest, called the odd
        ball.

        Your goal is to find out which one is the odd one.
        You are given a weighing scale!

        Good Luck and have fun!


Enter the number of balls for the game? 8

You are prompted to enter the balls
to be placed on the pans of the scale,
seperate each ball identifier with one
minimum space, e.g. 1 2 3
```

```
Enter the ball identifier(s) to be placed on left pan: 1 2  3
Enter the ball identifier(s) to be placed on right pan:  4 5

Your inputs for left:"1 2  3", right:" 4 5"

Invalid input!!!!

Please ensure correct ball identifiers (1-8)
are entered on each pan, no duplicate balls on either
or both pans.  Both pans should have the same number of
balls and must have at least one ball.
```

```
Enter the ball identifier(s) to be placed on left pan: 1 2 3
Enter the ball identifier(s) to be placed on right pan: 4 5 a

Your inputs for left:"1 2 3", right:"4 5 a"

Invalid input!!!!

Please ensure correct ball identifiers (1-8)
are entered on each pan, no duplicate balls on either
or both pans.  Both pans should have the same number of
balls and must have at least one ball.
```

```
Enter the ball identifier(s) to be placed on left pan:   1  2 3
Enter the ball identifier(s) to be placed on right pan: 4  5   6
The scale shows:  Left pan is down
Enter the odd ball number or press Enter to weigh: 3

Your answer is not correct!!!!


You are prompted to enter the balls
to be placed on the pans of the scale,
seperate each ball identifier with one
minimum space, e.g. 1 2 3

Enter the ball identifier(s) to be placed on left pan: 1
Enter the ball identifier(s) to be placed on right pan: 2
The scale shows:  Right pan is down
Enter the odd ball number or press Enter to weigh: 2

Congratulations!!!! Scale usage count:  2
```

MARKING CRITERIA

- Coding Styles – overall program structure including layout, comments, white spaces, naming convention, variables, indentation, functions with appropriate parameters and return.
- Program Correctness – whether or the program works 100% as per Scope.
- User Interaction – how informative and accurate information is exchanged between your program and the player.
- Readability counts – programs that are well structured and easy-to-follow using functions to breakdown complex problems into smaller cleaner generalized functions are preferred over a function embracing a complex logic with nested conditions and sub-functions! In other words, a design with clean architecture with high readability is the predilection for the course objectives over efficiency.
- KISS approach – Keep It Simple and Straightforward.
- Balance approach – you are not required to come up with a very optimized solution. However, take a balance between readability and efficiency with good use of program constructs.

ITEMS	PERCENTAGE	REMARKS
CODING STYLES	20%-25%	0% IF PROGRAM DOESN'T RUN
USER INTERFACE	15%-20%	0% IF PROGRAM DOESN'T RUN
FUNCTIONALITY	>55%	REFER TO SCOPE

DUE DATE

March 13rd, 2022, 11:59:59PM