CompSci 101 - Assignment 02

Due: 8:30am, 22nd January 2020.

Worth: This assignment is marked out of 30 and is worth 3% of your final mark.

Topics covered:

- Functions, if statements, loops

The work done on this assignment **must** be your own work. Think carefully about any problems you come across, and try to solve them yourself before you ask anyone else for help. Under no circumstances should you use code written by another person in your assignment solution or give your code to another student.

VERY IMPORTANT:

This assignment has three sections.
Section A is marked using Coderunner3 (20 marks). For this section you need to define twelve functions. The twelve functions, when inserted into the skeleton program, create the "one player against the computer" game of OhNo (see the game description below). Each function is described in Section A of this document and there is a Python skeleton program (`SkeletonA2.py`) which you **MUST** use to develop the twelve functions.

Section B (2 marks).
If the twelve functions from Section A are all correct then the Assignment two program should execute without error.

Section C (5 marks).
Section C of this assignment creates a totally automated game of OhNo.

Submission

Section A - Parts 1 - 12 are submitted using CodeRunner3.

Section B and Section C - Submit your completed Assignment 2 Python program (just one Python file) using the Assignment Dropbox:

https://adb.auckland.ac.nz/Home/

NOTES:

- This assignment is marked out of 30 and is worth 3% of your final mark. Three marks out of 30 are assigned for the style of your program (program docstring, good variable names, etc.).
- You must only use the features taught in CompSci 101.
- Example output using the completed program is available at the end of this document and also on the CompSci 101 Assignments website:
  https://www.cs.auckland.ac.nz/courses/compsci101ssc/assignments/

Assignment 2 – The game of OhNo

In this assignment, you will develop the OhNo game which is a dice throwing game. The game uses two dice and each player takes turns until one of the players reaches a total score of 50 or more. Initially both players start with 0 points.

Each turn, the player can choose to throw the two dice as many times as they like and add the total for their turn to their total score but, if at any throw:

- a six is thrown, the turn ends (and no points are added to the player's total score),
- two sixes are thrown (the turn ends and the player's total score reverts back to zero).

The program skeleton:

Some parts of the skeleton code for this assignment program are printed on Pages 7 and 8 of this document.

IMPORTANT NOTE: Initially, when you run the program, the program does very little (until you have completed several parts) because the functions used in the program have not yet been completed. As you complete each function, the program does more and more. Note that you will need to make changes to any code which is currently inside the skeleton functions.

Assignment 2 Section A (20 marks)

Download the Python skeleton program (`SkeletonA2.py`) from the CompSci 101 assignments website:

`https://www.cs.auckland.ac.nz/courses/compsci101ssc/assignments/`

and rename the `SkeletonA2.py` file to `"YourUsernameA2.py"`, e.g., `afer023A2.py`. Develop the solution to each of the twelve functions in the `SkeletonA2.py` program. Once you are happy that a function executes correctly, submit the whole function to CodeRunner:

https://coderunner3.auckland.ac.nz/moodle/

When you press the `Check` button in CodeRunner, you will receive immediate feedback telling you if you have passed all the tests for that function. You can submit as many times as you need. You need to submit each function separately. Your marks for Section A of Assignment 2 are obtained through CodeRunner3. When you have successfully completed all the functions and your code passes all the tests click the `"Finish attempt"` button.

In this version of the game, Player 1 is playing against the computer and the computer player 'chooses' to throw the dice once only each turn.

Some example output using the completed program is shown at the end of this document.

Some parts of the `main()`, the `play_a_game()` and `have_a_turn()` functions are shown on Pages 7 and 8 of this document. These three functions have already been defined for you and should not be changed in any way except for the changes indicated in Part 1 and Part 13.

You will need to change the name of the file to "YourUsernameA2.py", e.g., afer023A2.py. You are also welcome to change the players' names (or not).

1. welcome_to_game(player1_name, player2_name, target_score)

This function prints the following eight lines of output:

- A line of 60 stars,
- A string similar to:

```
'Welcome Adriana and AlwaysSaysNo to the game of "OhNo" by
afer023'
```

where the first name is the first parameter string, the second name is the second parameter string and the last string is your username.

- The string: `"First player to reach 50 is the winner."`, where the Number (50) is given by the third parameter.
- The following two lines of text:

```
In this version of the game, player 1 is playing against the computer and
the computer player 'chooses' to throw the dice once only each turn.
```

- A line of 60 stars,
- Two blank lines.

For example, the output produced by the following line of code is shown in the text box:

```
welcome_to_game("Adriana", "AlwaysSaysNo", 50)
```

```
************************************************************
Welcome Adriana and AlwaysSaysNo to the game of "OhNo" by afer023
First player to reach 50 is the winner.
In this version of the game, player 1 is playing against the computer and
the computer player 'chooses' to throw the dice once only each turn.
************************************************************
```

***** IMPORTANT*****

when checking this function in CodeRunner you **MUST** put `"by afer023"` (NOT your own username). This is so your output matches the expected output but in your actual Assignment 2 program put your own username (not afer023).

<---------------------------->

2. get_starting_player_number()

This function returns a random integer which is either the number 1 or the number 2. For example, the following line of code

```
print(get_starting_player_number(), get_starting_player_number())
```

prints a random combination of the number 1 and the number 2, e.g., `"1 2"`, `"2 1"`, `"1 1"` or `"2 1"`.

<---------------------------->

3. display_turn_info(player_name, total_score)

This function is passed two integer parameters: the player's name and the player's total score so far. The function prints two lines of output:

- A line of 60 stars,
- the player's name followed by the string, `"'s turn (Total score: "` followed by the total score and finally a closing parenthesis, `")"`.

For example, the output produced by the following line of code is shown in the text box:

```
display_turn_info("AlwaysSaysNo", 23)
```

```
**********************************************************
AlwaysSaysNo's turn (Total score: 23)
```

<---------------------------->

4. get_dice_roll()

This function returns one random number which is either a 1, 2, 3, 4, 5 or 6. For example, the following line of code

```
print(get_dice_roll(), get_dice_roll(), get_dice_roll())
```

prints three random digits, e.g., `"4 6 1"`.

<---------------------------->

5. display_dice(dice1, dice2)

This function is passed two integer parameters. The function prints two lines of output:

- One blank line,
- a line similar to: three blank spaces followed by `"You rolled 2 and 3 (5)"`

where the first number is given by the first parameter, the second number is given by the second parameter and the number inside the parentheses is the total of the two parameter numbers.

For example, the output produced by the following line of code is shown in the text box:

```
display_dice(2, 1)
```

```
   You rolled 2 and 1 (3)
```

<---------------------------->

6. check_for_double_six(dice1, dice2)

This function is passed two integer parameters. The function returns `True` if both the parameters are the number 6. Otherwise the function returns `False`. For example, the output of the following code is shown in the textbox below on the right:

```
print("1.", check_for_double_six(6, 6))
print("2.", check_for_double_six(3, 6))
```

<---------------------------->

```
1. True
2. False
```

7. check_for_a_single_six(dice1, dice2)

This function is passed two integer parameters. The function returns `True` if either one (or both) of the parameters is the number 6. Otherwise the function returns `False`. For example, the output of the following code is shown in the textbox below on the right:

```
print("1.", check_for_a_single(6, 1))
print("2.", check_for_a_single(5, 4))
```

<---------------------------->

```
1. True
2. False
```

8. get_user_choice(prompt)

This function is passed a string as a parameter. The function prompts the user for a response (using the parameter string as the prompt) and the function returns the **FIRST** character of the string entered by the user.

For example, the output of the following code is shown in the textbox below on the right. The user input is shown in bold.

```
result1 = get_user_choice("Enter yes or no: ")
result2 = get_user_choice("Enter ride or swim: ")
print("1.", result1)
print("2.", result2)
```

```
Enter yes or no: yes
Enter ride or swim: swim
1. y
2. s
```

<----------------------->

9. game_has_finished(player1_score, player2_score, target_score)

This function is passed three integer parameters: the scores of the two players and the number of points needed to win the game, `target_score`.

The function returns `True` if one of the players has a score which is equal to or greater than the `target_score`. Otherwise the function returns `False`. For example, the output of the following code is shown in the textbox below on the right.

```
result = game_has_finished(27, 76, 75)
print("1.", result)
print("2.", game_has_finished(50, 46, 50)
print("3.", game_has_finished(36, 46, 50))
```

```
1. True
2. True
3. False
```

<----------------------->

10. get_other_player_number(player_number)

This function is passed one integer parameter, the number of the current player (either 1 or 2). If the parameter is the number 1, the function returns the number 2. Otherwise the function returns the number 1. For example, the output of the following code is shown in the textbox below on the right.

```
result = get_other_player_number(2)
print("1.", result)
print("2.", get_other_player_number(2))
print("3.", get_other_player_number(1))
```

```
1. 1
2. 1
3. 2
```

<----------------------->

11. get_winner_name(player1_name, player2_name, player1_score, player2_score, target_score)

This function is passed five parameters: the name of the two players, the scores of the two players and the total number of points needed to win the game. The function returns the name of whichever of the two players has a score which is equal to or greater than the `target_score`:

For example, the output of the following code is shown in the textbox below:

```
name1 = get_winner_name("Anaru", "Hera", 36, 54, 50)
name2 = get_winner_name("Hine", "Kim", 55, 44, 55)
print("1.", name1, "2.", name2)
```

```
1. Hera 2. Hine
```

<----------------------->

12. display_game_results(winner_name, player1_name, player2_name, player1_score, player2_score)

This function is passed five parameters: the winner's name, the name of the two players and the scores of the two players. The function prints nine lines of output:

- Lines 1, 2, 7 and 8 are lines of 60 stars,
- Lines 3, 6 and 9 are blank lines,
- Line 4: starts with 3 spaces, followed by the winner's name followed by the string `" has won the game (Total score: "` followed by the score of the winning player and finally a closing parenthesis `")"`.
- Line 5: starts with 3 spaces, followed by the string `"Commiserations to "`, followed by the name of the non winning player, followed by the string `"(Total score: "` followed by the score of the non-winning player and finally a closing parenthesis `")"`.

For example, the output of the following code is shown in the textbox below:

```
display_game_results("Katete", "Katete", "Leo", 52, 39)
display_game_results("Ringo", "Ringo", "Dolly", 55, 49)
```

```
************************************************************
************************************************************

   Katete has won the game (Total score: 52)
   Commiserations to Leo (Total score: 39)

************************************************************
************************************************************

************************************************************
************************************************************

   Ringo has won the game (Total score: 55)
   Commiserations to Dolly (Total score: 49)

************************************************************
************************************************************
```

<----------------------->

Some parts of the skeleton program:

```
import random

def main():
    player1_is_automated = False  #SET THIS TO True WHEN YOU TEST STAGE 13
    player2_is_automated = True
    number_of_games = 1 #SET THIS TO 5 AFTER YOU HAVE COMPLETED STAGE 13
    printing_is_on = True
    #...
    player1_name = "Adriana"
    player2_name = "AlwaysSaysNo"
    target_score = 50
    welcome_to_game(player1_name, player2_name, target_score, printing_is_on)
    for i in range(number_of_games):
        winner_name = play_a_game(player1_name, player2_name, target_score,
                                  player1_is_automated, player2_is_automated)

def play_a_game(player1_name, player2_name, target_score, player1_is_automated,
                                      player2_is_automated, printing_is_on):
    player1_score = 0
    player2_score = 0
    game_has_ended = False
    current_player_num = get_starting_player_number()
    while not game_has_ended:
        if current_player_num == 1:
            player1_score = have_a_turn(player1_name, player1_score,
                                target_score, player2_score, player1_is_automated,
                                                                    False)
        else:
            player2_score = have_a_turn(player2_name, player2_score,
                                target_score, player1_score, player2_is_automated,
                                                                    True)
        if game_has_finished(player1_score, player2_score, target_score):
            game_has_ended = True
        else:
            current_player_num = get_other_player_number(current_player_num)

    winner_name = get_winner_name(player1_name, player2_name,
                                  player1_score, player2_score, target_score)
    display_game_results(winner_name, player1_name, player2_name,
                                                player1_score, player2_score)
    return winner_name
```

#1

#2

#9

#10

#11

#12

```
def have_a_turn(player_name, player_score, target_score,
          other_player_score, is_automated, always_chooses_no, printing_is_on):
    display_turn_info(player_name, player_score)                          #3
    score_this_turn = 0
    turn_has_finished = False
    number_of_rolls_this_turn = 0                                         #4
    while not turn_has_finished:
        dice1 = get_dice_roll()
        dice2 = get_dice_roll()
        number_of_rolls_this_turn += 1
        display_dice(dice1, dice2)                                        #5
        if check_for_double_six(dice1, dice2):                            #6
            score_this_turn = -player_score
            turn_has_finished = True
            print("\n   You rolled a double six.")
            print("   Your total score goes back to zero.", "(Total score: 0)")
        elif check_for_a_single_six(dice1, dice2):                        #7
            score_this_turn = 0
            turn_has_finished = True
            print("\n   You rolled a six,")
            print("   your score this turn is zero.", "(Total score: " +
                                                    str(player_score) + ")")
        else:
            score_this_turn = score_this_turn + dice1 + dice2
            print("   So far, your score this turn is", score_this_turn)
            if is_automated and always_chooses_no:
                user_choice = "n"
            elif is_automated:
                user_choice = get_automated_yes_no(score_this_turn,       #13
                                          player_score, other_player_score,
                                    number_of_rolls_this_turn, target_score)
            else:                                                         #8
                user_choice = get_user_choice("   Do you wish to roll again (y
                                                                  or n)? ")
            if user_choice == "n":
                turn_has_finished = True
                print("\n   Your score this turn is", score_this_turn, "(Total
                          score: " + str(player_score + score_this_turn) + ")")
    player_score = player_score + score_this_turn
    return player_score
```

Assignment 2 Section B (2 marks)

Make sure the completed twelve functions from Section A are all correct and your Assignment 2 program executes without error. Now you will be able to play the game of OhNo. You are welcome to change the names of the two players.

<---------------------------->

Assignment 2 Section C (5 marks)

13. get_automated_yes_no(so_far_this_turn, total_score, other_player_total_score, number_of_rolls_this_turn, target_score):

This last task is to automate Player 1's responses so that the user will no longer be required to input whether they wish to roll again or not. Before you start writing this function, you first need to change the value of the `player1_is_automated` variable to `True`, i.e., change the first line of the `main()` function from:

```
player1_is_automated = False  #SET THIS TO True WHEN YOU TEST STAGE 13
```

to

```
player1_is_automated = True
```

This function is passed five parameters: Player 1's current number of points for this turn, Player 1's current total score, the computer player's total points, the number of times the two dice have been rolled by Player 1 this turn and the number of points needed to win the game. This function needs to work out if Player 1 should roll again or not, i.e., the function should return the string `'y'` if Player 1 should roll again, otherwise the function should return the string `'n'`.

Some things you might consider when coding this function:

- How close is Player 1 to winning the game?
- Is the computer Player very close to winning the game?
- How many points has Player 1 already accumulated this turn?
- How many rolls of the dice has the player already had this turn?
- Any other considerations which will make the Player 1 strategy beat the 'never roll again' strategy.

Once you have tested that your program completes one game without error, you can now test your Part 13 function by running the program for many games and see if Player 1 wins more often than the computer player. To do this you need to change the value of the `number_of_games` variable to 500, i.e., change the third line of the `main()` function from:

```
number_of_games = 1 #SET THIS TO 500 AFTER YOU HAVE COMPLETED STAGE 13
```

to

```
number_of_games = 500
```

Note: once you have completed Part 13 which automates the Player 1 responses, if you have set the number of games to a number greater than 5, then the program will not print the output for individual games, i.e. , you will only see the final output (the number of games won by the winner and the number of games won by the runner up).

<----------------------------->

Example Output

Below is some example output produced by the completed program (not including the Part 13 function). In the example output the user input is shown in a bold pink font – your user input will not be coloured. Your program must execute in the way described and the output should have the same format as the output below:

```
************************************************************
Welcome Adriana and AlwaysSaysNo to the game of "OhNo" by afer023
First player to reach 50 is the winner.
In this version of the game, player 1 is playing against the computer and
the computer player 'chooses' to throw the dice once only each turn.
************************************************************


************************************************************
AlwaysSaysNo's turn (Total score: 0)

   You rolled 2 and 6 (8)

   You rolled a six,
   Your score this turn is zero. (Total score: 0)
************************************************************
Adriana's turn (Total score: 0)

   You rolled 1 and 2 (3)
   So far, your score this turn is 3
   Do you wish to roll again (y or n)? y

   You rolled 5 and 3 (8)
   So far, your score this turn is 11
   Do you wish to roll again (y or n)? yes

   You rolled 4 and 3 (7)
   So far, your score this turn is 18
   Do you wish to roll again (y or n)? y

   You rolled 4 and 4 (8)
   So far, your score this turn is 26
   Do you wish to roll again (y or n)? n

   Your score this turn is 26 (Total score: 26)
************************************************************
AlwaysSaysNo's turn (Total score: 0)

   You rolled 6 and 3 (9)

   You rolled a six,
   Your score this turn is zero. (Total score: 0)
************************************************************
Adriana's turn (Total score: 26)

   You rolled 6 and 1 (7)

   You rolled a six,
   Your score this turn is zero. (Total score: 26)
************************************************************
AlwaysSaysNo's turn (Total score: 0)

   You rolled 1 and 5 (6)
   So far, your score this turn is 6

   Your score this turn is 6 (Total score: 6)
************************************************************
Adriana's turn (Total score: 26)

   You rolled 2 and 4 (6)
   So far, your score this turn is 6
   Do you wish to roll again (y or n)? y
```

```
   You rolled 3 and 1 (4)
   So far, your score this turn is 10
   Do you wish to roll again (y or n)? y

   You rolled 4 and 2 (6)
   So far, your score this turn is 16
   Do you wish to roll again (y or n)? n

   Your score this turn is 16 (Total score: 42)
************************************************************
AlwaysSaysNo's turn (Total score: 6)

   You rolled 6 and 4 (10)

   You rolled a six,
   Your score this turn is zero. (Total score: 6)
************************************************************
Adriana's turn (Total score: 42)

   You rolled 1 and 6 (7)

   You rolled a six,
   Your score this turn is zero. (Total score: 42)
************************************************************
AlwaysSaysNo's turn (Total score: 6)

   You rolled 6 and 5 (11)

   You rolled a six,
   Your score this turn is zero. (Total score: 6)
************************************************************
Adriana's turn (Total score: 42)

   You rolled 4 and 2 (6)
   So far, your score this turn is 6
   Do you wish to roll again (y or n)? y

   You rolled 5 and 1 (6)
   So far, your score this turn is 12
   Do you wish to roll again (y or n)? no

   Your score this turn is 12 (Total score: 54)
************************************************************
************************************************************

   Adriana has won the game (Total score: 54)
   Commiserations to AlwaysSaysNo (Total score: 6)

************************************************************
************************************************************
```