ISTA 130: Fall 2015
Programming Assignment 5
(100 points)
While Loops and If Statements

Due: Thursday, October 8th by 11:59 pm
(submit via D2L dropbox)

Please read the instructions below carefully and follow them closely. All problems (and parts of problems) are required except as noted below.

Note: For any functions you are asked to write, you *must* use the *exact* function names given in the descriptions, and you *must* have parameters in the order shown in the description.

Note: to people who know what default values and/or keyword arguments are: do *not* use default values and do *not* use keyword arguments. If you don't know what these are it is unlikely you'll do it by accident. Ask us if you're not sure.

1 Preparation

Before you begin working on your program you'll need to learn a bit about the random module. In a Python shell:

- import the random module and do the following:

  - enter: help(random.randint) and read the documentation
  - enter: random.randint(0,1)

    * run that same statement many times until you know what it does
  - enter: random.randint(0,3)

    * run that same statement many times until you know what it does
  - enter: random.randint(4,6)

    * run that same statement many times until you know what it does
  - enter: random.randint(1,100)

    * What does it do?

- now enter: random.seed(100)

  - Next, 5 times in a row run this line: random.randint(0,10)
  - Each time write down the random number

- exit the Python shell, open a new shell, and enter: random.seed(100)

  - 5 times in a row run this line: random.randint(0,10)
  - Each time write down the random number
  - Compare to the numbers you already wrote down

- Random numbers are generated by a complicated mathematical process

  - by setting a *seed* value we can ensure that the exact same sequence of "random" numbers will be generated. This is very useful for testing code.

2 Dice Game: "Pig" (100 points)

For this program you will build a simple dice game called "Pig".

In this version of Pig, two players will alternate turns. Players each begin the game with a score of 0. During a turn a player will roll a six-sided die one or more times, summing up the resulting rolls. At the end of the player's turn the sum *for that turn* is added to the player's total game score.

If at any time a player rolls a 1, the player's turn immediately ends and he/she earns 0 points *for that turn* (i.e. nothing is added to the player's total game score). After every roll the player may choose to either end the turn, adding the sum from the current turn to his/her total game score, or roll again in an attempt to increase the sum.

The first player to 50 points wins the game.

2.1 Details

Open a text editor and create a new file called "pig.py". In the file:

1.) Write a function called print_scores that takes four parameters name1 (a string), score1 (an int), name2 (a string), score2 (an int). The function will:

   i.) **Print** the following message, using the current parameter values for the player names and scores (i.e. the value in name1 instead of Ziggy, the value in score1 instead of 18, etc):

      ---SCORES Ziggy: 18   Elmer: 23---

- Note: print a newline before this line
- Note: there is a single tab between SCORES and Ziggy
- Note: there is a single tab between 18 and Elmer
- Note: everywhere else use single spaces

   - Write code in main to test your function and verify it works correctly before moving on

2.) Write a function called check_for_winner that takes two parameters: name (a string) and score (an int):

   i.) if the score is 50 or more **print** the following message (obviously using the current parameter value for the name):

      THE WINNER IS: Ziggy!!!!!

- Note: use single spaces to separate words
- Note: there are 5 exclamation marks

   ii.) the function **returns** True if the score is 50 or more, otherwise it **returns** False

   - Write code in main to test your function and verify it works correctly before moving on (you may wish to comment out the code you used to test the previous function)

3.) Write a function called roll_again that takes a single string parameter to hold a player's name. The function will:

   i.) Enter a while loop in which it repeats the following steps:

     a.) Ask whether the player would like to roll again, using the following message:
     Roll again, Ziggy? (Y/N)

- Note: there is a single space after (Y/N)
- Note: use single spaces to separate words

     b.) If the player answers with "Y", "y", "N", or "n", exit from the loop

c.) Otherwise **print** the following message (replacing XXX with the user's entry):

I don't understand: "XXX". Please enter either "Y" or "N".

- Note: use single spaces to separate words

ii.) The function **returns** either True or False depending on whether the player wants to roll again

- Write code in main to test your function and verify it works correctly before moving on (you may wish to comment out the code you used to test the previous function)

4.) Write a function called play turn that takes a single string parameter to hold a player's name. The function will:

i.) **Print** the player's name as shown in the following example (again, obviously using the current parameter value for the name):

----------Ziggy's turn----------

- Note: there are 10 hyphens on either side of the message
- Note: use single spaces to separate words

ii.) The function then enters a while loop in which it repeats the following steps:

a.) generate a random integer between 1 and 6 inclusive
b.) **print** a message displaying the roll as shown in the following example:

    <<< Ziggy rolls a 4 >>>

- Note: there is a tab to the left of the first <
- Note: use single spaces to separate words

c.) if the roll is a 1:

- print the following message (replace Ziggy with the current name):

    !!! PIG! No points earned, sorry Ziggy !!!

  – Note: there is a tab to the left of the first !
  – Note: use single spaces to separate words

- set the player's current sum to 0
- make a change to ensure you exit the loop
- wait for the user to press enter before continuing. Use the following message (use single spaces to separate words): (enter to continue)

d.) if the roll is not a 1:

- add the roll to the player's current sum
- **print** the player's current sum as in the following example:
  Points: 12

4

- Note: there is a tab to the left of the word "Points"
- Note: otherwise use a single space
- use your roll again function to ask the player if he/she would like to roll again
- if the player would like to stop rolling, make a change to ensure you exit the loop

e.) finally **return** the number of points earned by the player for this turn

- Write code in main to test your function and verify it works correctly before moving on (you may wish to comment out the code you used to test the previous function)

5.) In main:

i.) Delete all of the code you've written to test the above functions and then write code to do the following:

ii.) Ask the user to enter an integer to use for the random seed:

Enter seed value:

iii.) Set the random seed to the value entered (e.g. random.seed(seed value))

iv.) **Print** two blank lines followed by the title of the game:

Pig Dice

v.) Ask the user to enter the first player's name

Enter name for player 1:

vi.) Ask the user to enter the second player's name

Enter name for player 2:

vii.) **Print** a greeting message in the following form:

Hello Ziggy and Elmer, welcome to Pig Dice!

- Note: there is a tab to the left of the word "Hello"
- Note: otherwise use single spaces

viii.) Initialize the players' scores to 0

ix.) **Print** their scores using your print scores function

x.) Enter a while loop in which you repeat the following steps:

a.) call your play turn function for player 1 and add any earned points to his/her total score

b.) print the players' scores (use your function)

c.) check whether player 1 won the game (use your function)

d.) if player 1 didn't win, call your play turn function for player 2 and add any earned points to his/her total

- print the players' scores

- check whether player 2 won the game

  e.) if either player won the game ensure that you exit the while loop

6.) Verify that your documentation makes sense and that you've added documentation to each of your functions.

7.) Verify that your program works properly

8.) Save your file (e.g. to your Locker)

9.) Upload your file to the Program 5 dropbox folder on D2L

2.2 Example Output

Below is an example of the output from running the finished program. If you enter 125 for the seed and make the *same* choices as the players do in this example your output should look exactly like this (note that spacing might not reproduce perfectly in this PDF, see instructions above for exact spacing used). User entered values are highlighted.

Enter seed value: 125

Pig Dice

Enter name for player 1: Ziggy

Enter name for player 2: Elmer

   Hello Ziggy and Elmer, welcome to Pig Dice!

--- SCORES Ziggy: 0   Elmer: 0 ---

---------- Ziggy's turn ----------

    <<< Ziggy rolls a 2 >>>

    Points: 2

Roll again, Ziggy? (Y/N) Y

  <<< Ziggy rolls a 2 >>>

    Points: 4

Roll again, Ziggy? (Y/N) Y

  <<< Ziggy rolls a 5 >>>

    Points: 9

Roll again, Ziggy? (Y/N) N

--- SCORES Ziggy: 9   Elmer: 0 ---

---------- Elmer's turn ----------

       <<< Elmer rolls a 3 >>>

       Points: 3

Roll again, Elmer? (Y/N) <mark>Y</mark>

     <<< Elmer rolls a 5 >>>

       Points: 8

Roll again, Elmer? (Y/N) <mark>Y</mark>

     <<< Elmer rolls a 6 >>>

       Points: 14

Roll again, Elmer? (Y/N) <mark>Y</mark>

     <<< Elmer rolls a 2 >>>

       Points: 16

Roll again, Elmer? (Y/N) <mark>N</mark>

--- SCORES Ziggy: 9   Elmer: 16 ---

---------- Ziggy's turn ----------

     <<< Ziggy rolls a 3 >>>

       Points: 3

Roll again, Ziggy? (Y/N) <mark>Y</mark>

     <<< Ziggy rolls a 2 >>>

       Points: 5

Roll again, Ziggy? (Y/N) <mark>Y</mark>

     <<< Ziggy rolls a 5 >>>

       Points: 10

Roll again, Ziggy? (Y/N) <mark>Y</mark>

     <<< Ziggy rolls a 3 >>>

       Points: 13

Roll again, Ziggy? (Y/N) <mark>Y</mark>

     <<< Ziggy rolls a 5 >>>

       Points: 18

Roll again, Ziggy? (Y/N) <mark>N</mark>

--- SCORES Ziggy: 27   Elmer: 16 ---

---------- Elmer's turn ----------

```
        <<< Elmer rolls a 6 >>>
            Points: 6
Roll again, Elmer? (Y/N) Y
        <<< Elmer rolls a 1 >>>
            !!! PIG! No points earned, sorry Elmer !!!
(enter to continue)


--- SCORES Ziggy: 27        Elmer: 16 ---
---------- Ziggy's turn ----------
        <<< Ziggy rolls a 1 >>>
            !!! PIG! No points earned, sorry Ziggy !!!
(enter to continue)


--- SCORES Ziggy: 27        Elmer: 16 ---
---------- Elmer's turn ----------
        <<< Elmer rolls a 6 >>>
            Points: 6
Roll again, Elmer? (Y/N) N


--- SCORES Ziggy: 27        Elmer: 22 ---
---------- Ziggy's turn ----------
        <<< Ziggy rolls a 6 >>>
            Points: 6
Roll again, Ziggy? (Y/N) Y
        <<< Ziggy rolls a 2 >>>
            Points: 8
Roll again, Ziggy? (Y/N) Y
        <<< Ziggy rolls a 4 >>>
            Points: 12
Roll again, Ziggy? (Y/N) Y
        <<< Ziggy rolls a 5 >>>
            Points: 17
```

and so on...