Zelle 3e Chapter 9 Coding Assignment

General Instructions

My expectations for your work on coding assignment exercises will grow as we progress through the course. In addition to applying any new programming techniques that have been covered in the current chapter, I will be expecting you to follow all of the good programming practices that we have adopted in the preceding weeks. Here is a quick summary of good practices that we have covered so far:

- Include a single-line comment with name of program file.
- Include a single-line comment that describes the intent of the program.
- Place your highest-level code in a function named `main`.
- Your code should be factored such that there is a function in your program for each part of the problem.
- Each function should contain code relating to the same thing – it should have *high cohesion*.
- Functions should know as little as possible about the workings of other functions – they should have *low coupling*.
- Include a final line of code in the program that executes the `main` function.
- Follow all PEP-8 Python coding style guidelines enforced by the PyCharm Editor. For example, place two blank lines between the code making up a function and the code surrounding that function.
- Output printed by the program (both prompts and results) should be polite and descriptive.
- Choose names for your variables that are properly descriptive.
- Choose names for your functions that are properly descriptive.
- Close all files before the conclusion of the program.
- Model your solution after the code that I demonstrate in the tutorial videos.
- Remember to test your program thoroughly before submitting your work.
- Your code must pass all relevant test cases. Make sure that it passes tests at the boundaries created by *if, else*, and *elif* conditions in your program (boundary value tests).

Assignment Overview

The goal of this assignment is to build a popular children's dice game named "Beat That". Beat That is an educational game in which children learn strategic thinking and the concept of place value. See http://www.anaheimymca.org/wp-content/STEM%20Beat%20That!%20Math%20Dice%20Game.pdf .

While the rules of the game can be flexible, we will be playing a basic version in which 2 players (Player A and Player B) are rolling 2 dice in a 5-round game.

Game play is based on rounds. In each round, players take a turn in which they roll 2 dice. After rolling the dice, the player decides how to use the two dice values to form a 2-digit number. The goal is to arrange the digits so that they form the highest number possible. Since we are building our computer programming skills rather than our number literacy skills, our program will automatically make the choice for the player – choosing the highest of the two possible combinations as the player's number choice for that round.

At the close of the round, our program will compare players' number choices for that round to determine the round scoring. The player with the highest number choice wins the round. There are 3 possible scoring outcomes of a round:

- The Players Tie this Round
- Player A Wins this Round
- Player B Wins this Round

The program will print a message to announce the outcome of the round.

The player who wins a round scores 1 point in the overall game. The player who loses the round does not score a point in the overall game. If the players are tied for the round, then neither player scores a point in the overall game.

When 5 rounds have been played, the program will determine the outcome of the overall game based upon points earned. There are 3 possible outcomes of a game:

- The Players Tie the Game
- Player A Wins the Game
- Player B Wins the Game

The program will print a message to announce the outcome of the game.

Our strategy for developing the Beat That game is to do it in two stages. Rather than designing and coding a solution to the overall problem in one attempt, we will look for a useful subset of the functionality to be our first development goal.

The subset that we have chosen as our first development goal is to develop a program that plays one round of Beat That. The program will be called *beat_that_single_round*. Rounds are a natural breaking point in the game. While playing one round will not be as satisfying as playing a whole game, playing one round does yield some player satisfaction. So, it represents a good choice for the goal of our first development stage.

When our first development goal has been reached, we will undertake the final goal, developing a 5-round Beat That game. The program will be called *beat_that_five_round_game*. This version will include all of the functionality from the prior version plus scorekeeping and winner determination for a 5-round game. The design for the later program should include the design from the earlier program. Likewise, the code for the later program should include as many parts as possible from the earlier program. Please note that there is no formal relationship between these two programs in a system architecture sense. The later program does not call functions within the earlier program. Rather, the earlier program will be cannibalized, and its parts used in the creation of the later program.

Please see the following section for details on the 2 exercises that make up this assignment.

Exercise 1
Create a program named *beat_that_single_round.* Based upon the description above, the program should implement the functionality for playing a single round of Beat That.

When coding and testing this program, follow the approach that I take in the first part of the tutorial video where I design and build the program that simulates one racquetball game.

When this program is run, the console sessions should look like this:

```
Playing one round of Beat That...
Player A rolls (5, 4)
Player A chooses 54
Player B rolls (2, 1)
Player B chooses 21
Player A Wins This Round!


Playing one round of Beat That...
Player A rolls (4, 6)
Player A chooses 64
Player B rolls (2, 2)
Player B chooses 22
Player A Wins This Round!


Playing one round of Beat That...
Player A rolls (5, 1)
Player A chooses 51
Player B rolls (1, 5)
Player B chooses 51
Players Tie This Round.
```

Exercise 2
Create a program named *beat_that_five_round_game*. Based upon the description above, the program should implement the functionality for playing a 5-round game of Beat That.

When coding and testing this program, follow the approach that I take in the second part of the tutorial video where I design and build a program that simulates many racquetball games.

When this program is run, the console session should look like this:

```
Playing round 1 of Beat That...
Player A rolls (1, 5)
Player A chooses 51
Player B rolls (5, 6)
Player B chooses 65
Player B Wins Round 1

Playing round 2 of Beat That...
Player A rolls (1, 4)
Player A chooses 41
Player B rolls (1, 2)
Player B chooses 21
Player A Wins Round 2

Playing round 3 of Beat That...
Player A rolls (1, 2)
Player A chooses 21
Player B rolls (5, 1)
Player B chooses 51
Player B Wins Round 3

Playing round 4 of Beat That...
Player A rolls (4, 1)
Player A chooses 41
Player B rolls (5, 1)
Player B chooses 51
Player B Wins Round 4

Playing round 5 of Beat That...
Player A rolls (4, 5)
Player A chooses 54
Player B rolls (5, 6)
Player B chooses 65
Player B Wins Round 5

Game score:
Player A has won 1 rounds.
Player B has won 4 rounds.
Player B Wins This Game
```

Tools
Use PyCharm to create and test both python programs.

Submission Method
Follow the process that I demonstrated in the tutorial video on submitting your work. This involves:

- Locating the properly named directory associated with your project in the file system.
- Compressing that directory into a single .ZIP file using a utility program.
- Submitting the properly named zip file to the submission activity for this assignment.

File and Directory Naming
Please name your Python program files as instructed in each exercise. Please use the following naming scheme for naming your project:

YourLastName_YourFirstName_exercises_zelle_3e_chapter_09

When you have compressed your project directory into a .ZIP file, it should have the following name structure:

YourLastName_YourFirstName_exercises_zelle_3e_chapter_09.zip

Due By
Please submit this assignment by the date and time shown in the Weekly Schedule.