Week 7 Review

Submit Assignment

Due Mar 1 by 11:59pm **Points** 21 **Submitting** a text entry box or a file upload

Overview

When attempting these problems, please keep in my mind the nature of **Academic Honesty** in this course. This week you should submit your response to the Week 7 Review. This will support your review of the topics covered this week and prepare you for writing your program.

Programming Exercises

The content of this week's module is less about new expression and algorithmic structures, and more about how you design and build your programs. Although correctly meeting problem requirements remains part of how you will be graded, we will now also focus on *how* you designed the program that meets those requirements. Specifically, we will look for:

- appropriate separation of concerns
- appropriate use of function interfaces (i.e., using parameters and return values, not using variables declared as part of a script to make them accessible in all function scopes)
- comments explaining the why behind more difficult areas of code
- good function, parameter, and variable names that communicate the purpose and make the program more readable

Please keep in mind there is no single correct answer for *how* you design the program: there are many good approaches that are worth full credit. If you find yourself in a spiral of trying to perfectly optimize function interfaces and writing a lot of functions with a single line of code in them, you are overthinking it.

1. **[roll_until_sum.py]** Write a program that simulates rolling two 6-sided dice, and repeatedly rolls them until a target sum is reached. A user should input what the target sum is. The program should output the result of rolling each die and the sum, finally reporting how many rolls it took to reach the target sum. For example:

```
This program rolls two 6-sided dice until their sum is a given target value.
Enter the target sum to roll for: 9

Roll: 5 and 3, sum is 8
Roll: 3 and 1, sum is 4
Roll: 4 and 1, sum is 5
Roll: 4 and 1, sum is 5
```

```
Roll: 6 and 3, sum is 9
Got it in 5 rolls!
```

2. **[random_shapes.py]** You will write a program that generates random drawings that can be drawn using the **shape painter**.

The shape painter consumes a drawing file where each line corresponds to a rectangle or circle shape that is part of the drawing. A line in a drawing file has the following structures:

```
Circle; 80, 72; 15; 32, 208, 86
Rectangle; 145, 106; 421, 274; 32, 208, 205
```

The first line would draw a circle with a center point at coordinate (80, 72), a radius of 15, and a fill color with 32 for the red intensity, 208 for the green intensity, and 86 for the blue intensity. The second line would draw a rectangle with an upper left corner at coordinate (145, 106), a lower right corner at coordinate (421, 274), and a fill color with 32 for the red intensity, 208 for the green intensity, and 205 for the blue intensity.

Your goal is to create a program that generates drawing files by randomly creating these shape lines. A user should input, in this order

- the name of the file to write the shape lines out to
- the number of shapes to generate and write to the file as lines.

For each line to be written into the drawing output file, your program should randomly choose:

- the shape type
- the corresponding Point coordinates: currently the shape painter assumes the window size is 500 pixels wide by 500 pixels tall so you should only generate Point coordinate values that will remain within those bounds
- a fill color: you should not randomly generate each intensity color for the full range of 0 to 255. In order to make your drawings more attractive, you should use smaller ranges of intensity values, and those smaller ranges are up to you. For example, you could randomly select blue intensity values in the range 192 to 255.
- any other necessary values (e.g., the radius of a Circle).

An example interaction:

```
Enter the drawing file name to create: d3.txt
Enter the number of shapes to make: 30
```

This will write 30 shape lines to the file `d3.txt`. A sampling of those lines:

```
Rectangle; 127, 143; 318, 332; 32, 208, 253
Circle; 302, 323; 15; 32, 208, 255
Circle; 458, 253; 47; 32, 208, 129
Circle; 80, 72; 15; 32, 208, 86
Rectangle; 145, 106; 421, 274; 32, 208, 205
```

The user should then be able to open that file with the included `shape_painter.py` program and see what the drawing looks like.

Submission

Please post all necessary .py files to Canvas and include your answers to the questions under the "Canvas Submission" banner in the textbox provided.

Canvas Submission

When you submit this assignment here in Canvas, I would like you to answer the following question(s):

1. How many hours do you estimate you used completing this assignment?
2. What was easiest for you when completing this assignment?
3. What was the most difficult challenge you experienced when completing this assignment?

Week 7 Review Rubric (1)

Criteria	Ratings			Pts
1. Roll Until Sum: Input Sum	**2.0 pts** **Full Marks**	**0.0 pts** **No Marks**		2.0 pts
1. Roll Until Sum: Correctly Simulates Dice	**1.0 pts** **Full Marks**	**0.0 pts** **No Marks**		1.0 pts
1. Roll Until Sum: Outputs Rolls and Sums	**2.0 pts** **Full Marks**	**0.0 pts** **No Marks**		2.0 pts
1. Roll Until Sum: Ends on Input Sum	**1.0 pts** **Full Marks**	**0.0 pts** **No Marks**		1.0 pts
2. Random Drawing: Randomly Chooses Shapes	**2.0 pts** **Full Marks**	**0.0 pts** **No Marks**		2.0 pts
2. Random Drawing: Points Within Window Bounds	**2.0 pts** **Full Marks**	**0.0 pts** **No Marks**		2.0 pts
2. Random Drawing: Color Intensity Values Valid	**2.0 pts** **Full Marks**	**0.0 pts** **No Marks**		2.0 pts
2. Random Drawing: Outputs File Readable by Shape Painter	**4.0 pts** **Full Marks**	**2.0 pts** **Minor mistakes making file unreadable**	**0.0 pts** **No Marks**	4.0 pts
2. Random Drawing: Appropriately Separates Concerns	**1.0 pts** **Full Marks**	**0.0 pts** **No Marks**		1.0 pts
2. Random Drawing: Uses Function Appropriately	**2.0 pts** **Full Marks**	**0.0 pts** **No Marks**		2.0 pts
2. Random Drawing: Code is Readable due to Naming and Comments	**1.0 pts** **Full Marks**	**0.0 pts** **No Marks**		1.0 pts

Criteria	Ratings		Pts
Answers Questions in Canvas Submission	**1.0 pts** **Full Marks**	**0.0 pts** **No Marks**	1.0 pts
Total Points: 21.0			