

IT 212 Final Exam

Spring 2021

If you haven't already, first complete the quiz Final Questions on D2L.

You may use a computer to complete these problems and questions. Paste your answers into a text file called final.txt. You may refer to notes and online resources, but you must complete the problems without help from any person. Always submit your best effort, even if the code doesn't fully work.

When providing your answers, you are not required to show that your code works.

1. Write an interactive Python function called **super_greet**. It doesn't take any parameters, nor does it return any value. This function should prompt the user for their name and then read it in as a string (use input). It should then respond with the greeting, "Hello there, [name]!" where [name] is the name the person typed in. In addition, if the name is longer than 8 characters, it should add a second line: "That's a long name!". Note that super_greet does not take any parameters, nor does it return any values. (6 points)

Below are examples of how the function should run:

```
>>> super_greet()
What is your name? Sam
Hello there, Sam!
```

```
>>> super_greet()
What is your name? Bartholomew
Hello there, Bartholomew!
That's a long name!
```

2. This question involves the use of a Dice class, which is defined in the folder dice. Also included is a file demo.py that demonstrates how the Dice class can be used. After you have studied the code, write a new Python function called **roll_count** that uses the Dice class. This function has two parameters, both integers. The first parameter indicates how many times that the dice are rolled. The second parameter indicates which roll should be counted. For example, if the second parameter is 4, it should count the number of times that 4 is rolled. The function should then return the count.

Your function should use the Dice class and its methods to roll the dice. In other words, the code you write should not directly call any random functions. Also, you do not need to modify the Dice class.

Below are some sample runs:

```
>>> roll_count(1000, 7)
189
>>> roll_count(1000, 7)
170
>>> roll_count(1000, 7)
159
>>> roll_count(1000, 2)
38
>>>
```

Note that the function doesn't always return the same value but generally produces a higher count for 7 than 2, because there are more ways to roll a 7 on a pair of dice. (12 points)

3. Refer to the Dice class used in the previous problem. The following lines of Python code use the Dice class (which you can also find in demo.py):

```
dice_set = Dice()
print(dice_set)

ds2 = Dice()
dice_set.roll_again()

if dice_set == ds2:
    print("Lucky!")
else:
    print("Not so lucky!")
```

- a) Explain how the above code works. Your explanation must include which Dice methods are called and what happens conceptually in each method. Your answer shouldn't predict any output (not possible anyway since it's random). (6 points)
- b) Note that the Dice class only works for a pair of dice. Explain what would need to be modified in the Dice class so that when a Dice object is created, the programmer could specify the number of dice that are rolled. For example, the programmer may write the following code:

```
dice_set3 = Dice(3)
```

This would then create an object of three dice and roll all three to create a sum.

Your answer doesn't need to create any code. Explaining what would need to change is fine. (6 points)

4. Write a Java function called **greeting_repeat**. It should take two parameters. The first parameter is a string representing a person's name. The second parameter should be an integer. When called, `greeting_repeat` should write "Hello [name]!" repeatedly with the number of times equal to the second parameter. The function does not return anything (declared as void). For example, consider this statement that could run in the main function of a Java class:

```
greeting_repeat("Mary", 4);
```

It would produce the following output:

```
Hello Mary!  
Hello Mary!  
Hello Mary!  
Hello Mary!
```

After you have tested your function, write additional code that queries the user for their name. Then, run `greeting_repeat` to greet this person 10 times. Include all of your Java code in the `final.txt` file. (12 points)