# Function Practice: Printing a Range of Numbers, *now with parameters!*

**The goal for this exercise** is to make sure that you can create and use basic functions, and that you can pass information to those functions using parameters/arguments, and get return values back from those functions. For this exercise, you are intended to do a slightly more open-ended exercise that utilizes what you've learned previously.

In the starter project, you will find Testing\_Randomness.RunExercise method, already filled in. You need to implement the getAverage method and the separate checkAverage methods on a separate HelperClass so that Testing\_Randomness.RunExercise (1) accomplishes everything described here, and (2) makes good use of the two methods that are used in the (commented-out) starter code.

Note: You may have to create the HelperClass (and any necessary methods) in order to make this work.

For this exercise, let's imagine that you suspect that the random number generator isn't really all the random. Specifically, you think that if you generated 1,000 random numbers in the range of 1 to 6 that your 1,000 numbers would be skewed in some fashion (either they tend to be high, or they tend to be low). You propose to test this by generating 1,000 numbers that range in value from 1 to 6 (including, possibly, both 1 and 6), calculate the average of all those numbers, and then compare the average of your random numbers to the mathematically calculated midpoint of the range (in this case, the mathematically calculated midpoint should be sum of the lowest possible number and the highest possible number, divided by 2.

However, you're not going to be satisfied with just checking random numbers between 1 and 6. You want to repeat this experiment with the ranges 1through 10, and then again with 1 through 20.

(As you can see in the provided starter code, you will need to separate out the two steps – generating the random numbers and calculating the average of them, and then separately printing out a message about whether the calculated average is the same as the midpoint or not).

Here is some sample output from the program (because the program uses random numbers, your output may feature different numbers (and even different conclusions (messages), but should follow this basic format):

The average random number (range: 1 to 6) was expected to be 3.5, but was actually 3.532

The average random number (range: 1 to 10) was expected to be 5.5, but was actually 5.57

The average random number (range: 1 to 20) is 10.5, as expected

**What you need to do for this exercise:**

1. In the starter project, uncomment the Testing\_Randomness.RunExercise method, and fill in both that and the other methods on the HelperClass so that they behave as described above and the class overall accomplishes it's goals.