Alan, Yazmin, Betsy
Random Numbers
CSC 2300
4.2.17

April 05, 2017

Sample output:

Part 1 Histogram:
*** = approximately 40 occurrences**

```
 1: *
 2: **
 3: ****
 4: ******
 5: **********
 6: ***************
 7: ********************
 8: *************************
 9: ****************************
10: ******************************
11: ************************
12: ************************
13: ********************
14: ***************
15: **********
16: *******
17: *****
18: ***
19: *
20: *
```

Part 2 Histogram:
*** = approximately 72 occurrences**

```
 1:
 2:
 3:
 4: *
 5: *
 6: ***
 7: ********
 8: ***************
 9: ****************************
10: ******************************
11: ************************
12: ******************
13: *********
14: ****
15: *
16: *
17: *
18:
19:
20:
```

Part 3 Histogram:
*** = approximately 61 occurrences**

```
 1:
 2: *
 3: *
 4: ***
 5: *******
 6: **************
 7: ********************
 8: ****************************
 9: ******************************
10: *************************
11: ******************
12: **********
13: *****
14: **
15: *
16: *
17:
18:
19:
20:
```

Analysis/Explanation: When generating 3 numbers the histograms is shown as a bell curve. It seems that 10 is the mean to most of the generated histograms while the min is 1 and the max is 20 with an occurrence of 37 to 40. Since 3 numbers are being generated, there is a high probability that the average will range from 1 to 20. Because 3 is a small digit, there is a higher chance the average will be spread apart. It will peak near the middle because there are many ways to have an average around that area, while the ends will occur less since there will be less combinations to have an average specifically to those end numbers. For example to get an average of 1 the 3 integers must only be all 1s and for an average of 20 the integers must only be all 20s. This occurrence will surely be rarely randomized.

In the second part of the project, the histogram graph is also a bell curve. The bell curve is a lot steeper, ranging from 4 to 16 with 70-72 occurrences per asterisk. The curve gets steeper because when generating 10 numbers, there is less of a chance that you will get all low numbers just as there is less of a chance that you will get all very high number.

For the third part of the project, ten random numbers were generated again between 1 and 20 inclusive, but the chance of getting a particular number was not equally likely. In this part, there is a 50% chance to generate a number between 1 and 5 inclusive and a 50% chance to generate a number between 11 and 20 inclusive. Looking at the results, the most frequent average was 9 with a runner-up of 8, and the histogram ranges with averages from 2 to 16 inclusive. Since there is an equal chance to generate a number from the range 1-5 or 11-20 and the range 1-5 is much more compact and lower than the range, 11-20, the overall average should be less than 10 and shift the bell curve favorably to the left, or lower averages. To analyze further, using the fact that there is an equal chance to generate a number in both ranges, the mean one may get with the range 1-5 is 2.5, while the mean one may get with the

range 11-20 is 15. When taking the average of the means in these ranges, the overall average is 8.75, which justifies the overall average of 9 shown in the histogram with the runner-up of 8.

Source Code:

Part 1:

```
import support.Histogram;

import java.util.Random;

public class RandomNumbers {

	public static void main(String[] args) {

	final float N = 10000;

	Random rand = new Random();

	final int rndNum = 3;

	Histogram graph = new Histogram(1, 20);

	for (int a = 0; a < N; a++) {

				int sum = 0; // resetting the sum

				for (int i = 0; i < rndNum; i++) {

				int n = rand.nextInt(20) + 1;

				sum += n;

		}

				int average = sum / rndNum;

				graph.submit(average);

	}

	System.out.println(graph);

	}

}
```

Part 2:

```
import support.Histogram;

import java.util.Random;

public class RandomMeansPt2
{

	public static void main(String[] args)
	{
		final float N = 10000;
		final int flips =10;
		Random rand = new Random();

		Histogram graph = new Histogram(1, 20);

		for (int a = 0; a < N; a++)
		{
			int sum=0;
			for (int i = 0; i < flips; i++)
			{
				int n = rand.nextInt(20) + 1;
				sum += n;
			}
			int average = sum / flips;
			//System.out.println(average);
			graph.submit(average);
		}
		System.out.println("Graph 2" +graph);


	}

}
```

Alan, Yazmin, Betsy
Random Numbers
CSC 2300
4.2.17

Part 3

```
import support.Histogram;
import support.MultiDie;
import java.util.Random;
public class RandomNumbers {
	public static void main(String[] args) {
	final float N = 10000;
	final int randomNums = 10;
	Random rand = new Random();
	MultiDie coin = new MultiDie(2);

	Histogram graph = new Histogram(1, 20);


	for (int a = 0; a < N; a++) {
		int sum=0; //resetting the sum
		for (int i = 0; i < randomNums; i++) {
		coin.roll();
		int n;
		if (coin.getFaceValue()==1)
		n = rand.nextInt(5) + 1;
		else
		n = rand.nextInt(10) + 11;
		//		System.out.println("Number is :" + n);
		sum += n;
		}
		int average = sum / randomNums;

	//		System.out.println("sum is: " + sum);
	//System.out.println("Average is: " + average);

	graph.submit(average);
	}
	System.out.println(graph);
	}
}
```