Assignment #2 – Due January 14

**Algorithms**

**Description:** The assignment can be done **individually** or in teams of **two**. Submit one assignment per team of two via Omnivox and NOT MIO. Assignments sent via MIO will be deducted marks. Assignments must be done alone or in groups and collaboration between individuals or groups is strictly forbidden. Submissions that are late will receive a 10% penalty per day, and after four days of no submission a mark of zero will be given.

**General Guidelines When Writing Your Programs:**

Include the following in comments at the top of your program:

// -----------------------------------------------------

// Assignment (include number)

// Written by: (include your name(s) and student ID(s))

// Short Description of your project/code and how you designed it.

// -----------------------------------------------------

Throughout your program, include comments in your program describing the main steps where necessary.

**QUESTION ONE: Random Number Generator (20 points)**

**Purpose:** The purpose of this assignment is to have you experience manipulating arrays and creating a class and using it in a driver program.

**Teams:** The assignment can be done individually or in teams of 2. Submit one assignment per team; be sure to have both team members’ name in the comments at the top of the assignment.

Write a program to simulate the rolling of two dice. The program should use **Math.Random** and **util.Random** object to roll the first and second die. Using a two dimensional array to tally the number of times each sum appears using the random number generator methods.

Your program should:

1. Display a welcoming message

2. Ask the user how many sides each die has

3. Create the 2-d array to hold the frequency counter of each possible sums

4. Ask the user how many times they want the dice to be rolled using both methods

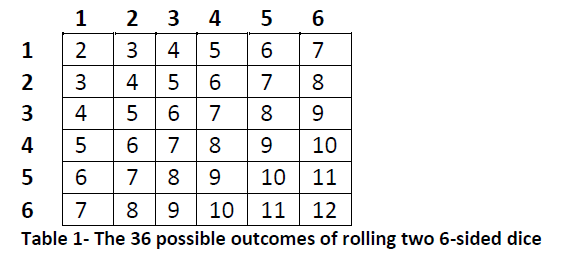
5. Print the expected frequencies and actual frequencies and percentages each sum appeared in both methods in a tabular format.

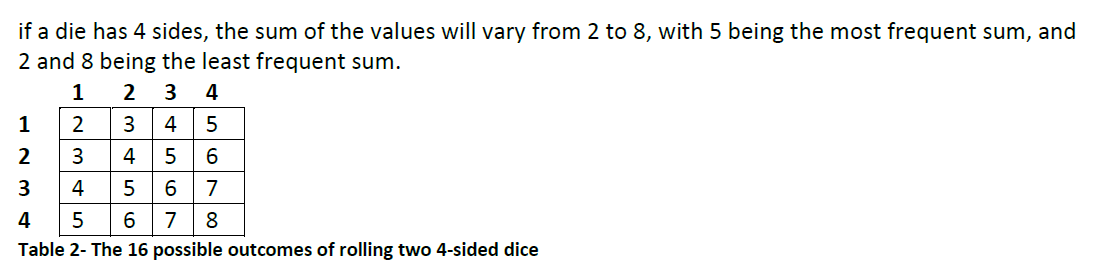
6. Repeat steps 3 to 5 as long as the user wants.

7. End with a closing message

See the sample output for the output format. Your percentages must be displayed with 2 places after the decimal. Use *printf* to format your output.

**Note:** if a die has 6 sides, the sum of the values will vary from 2 to 12, with 7 being the most frequent sum, and 2 and 12 being the least frequent sum. For example:





Additional Resources & Readings:

<https://stackoverflow.com/questions/738629/math-random-versus-random-nextintint>

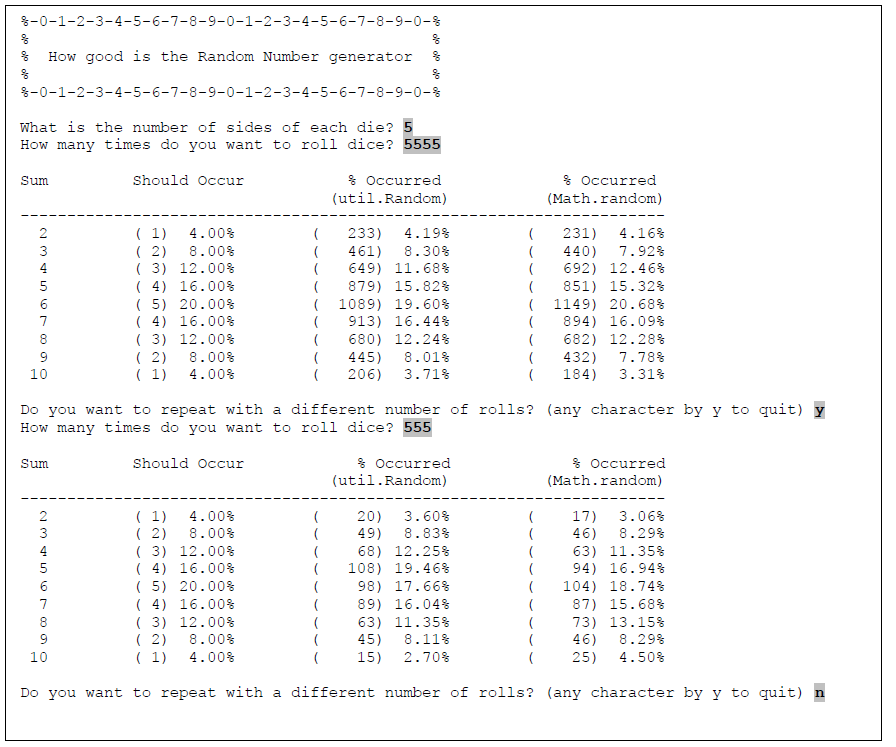
<https://dzone.com/articles/random-number-generation-in-java>

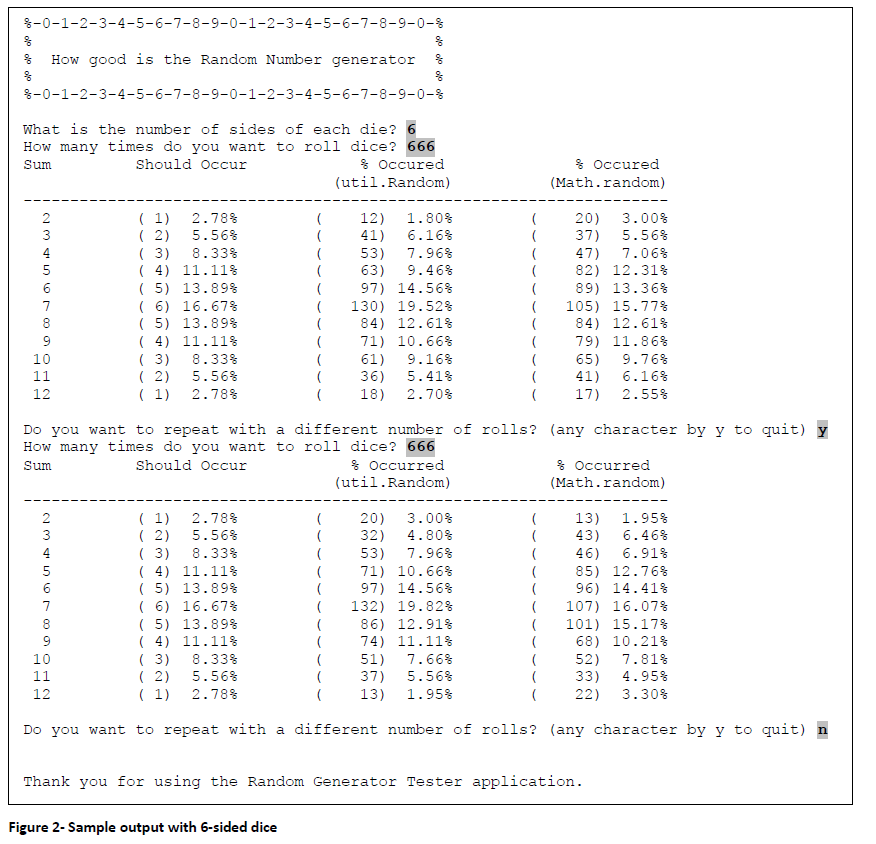
<http://www.fredosaurus.com/notes-java/summaries/summary-random.html>

<https://docs.oracle.com/javase/10/docs/api/java/util/Random.html>

<https://docs.oracle.com/javase/7/docs/api/java/lang/Math.html>

The following are 2 sample output screens. User input is highlighted and bolded

**Figure 1- Sample output with 5-sided dice**



Grading Grid:

|  |  |  |
| --- | --- | --- |
| **Criteria for Questions 1 – (1 pts.)** | | |
| Comments & Programming Style which includes:  - Description of the program (authors, date, purpose)  - Significant names for variables  - Description of variables  - Description of the algorithm | 1 | pts. |
| **Question 1 – (19 pts).** | | |
| No bugs occur | 4 | pts. |
| Menu operation/error checking | 1 | pt. |
| Clarity of output | 2 | pts. |
| Column 1 (Should Occur) | 2 | pts. |
| Column 2 (util.Random) | 3 | pts. |
| Column 3 (Math.Random) | 3 | pts |
| Correct Probabilities | 3 | pts |
| Program loops | 1 | pts |
| **Total 20 pts** | | |

Submission:

You are allowed to work on a team of 2 students at most (including yourself!). Any teams of 3 or more students will result in 0 marks for all team members. If your work on a team, ONLY one copy of the assignment is to be submitted for both members. Zip together the source codes. (Please use WINZIP or WINRAR).

The zip file should be called a#\_studentID, where # is the number of the assignment studentID is your student ID(s) number. For example, for the first assignment, student 1234567 would submit a zip file named a1\_12345678.zip. If you work on a team and your IDs are 12345678 and 34567890, you would submit a zip file named a1\_1234567\_4567890.zip

Submit only ONE version of an assignment. If more than one version is submitted, the latest (most recent) version submitted will be the only one considered.