Due: September 6, Thursday, **5:00 pm** (zero credit if late turn-in)

Write two Java programs as follows:

- Use "default" package: i.e., Your program should not have "package ...." on the top of the program.
- Define a public class named **`MyDice`** as follows:
  - It has states to remember the results of dice rolling (i.e., the # of 1's, the # of 2's, … the # of 6's)
  - It has a command to roll the dice N times
    - Study **`Math.random( )`** API and use it to generate random numbers. Do not use **`Random`** class.
  - It has a query to get the results of dice rolling.
  - This class has strict queries and commands only. Do NOT use any side effects of the methods. Significant penalty will be applied if you use any side effects. This means you must not use printf( ) or Scanner in this class.
- Define a public class named **Quiz04** as follows for the purpose of testing **MyDice** and displaying a table as shown in the sample run below:
  - Create objects of MyDice.
  - Using commands and queries for the objects, display the following tables:
    - Prompt the user for an integer and validate the input if the range is 1,000 <= input <= 100,000. You may assume the user always enters an integer.
    - Re-prompt if validation fails.
  - Display the results of dice rolling simulation as follows:

```
The dice was rolled XXXXX times.
  the number of 1s = XX
  the number of 2s = XX
  ...
  the number of 6s = XX
```

Note: You will observe that as the number of dice rolling becomes larger, the random numbers are evenly distributed among 1 through 6.

Turn-in: Submit programs via D2L (Dropbox -> Quiz04 link) as follows (**Do NOT submit them individually**):

1. Create a folder named **quiz04**. Do this outside NetBeans.
2. Copy Quiz04.java,MyDice.java into quiz04 folder.
3. Compress quiz04 folder. (You can choose a compress utility by right-clicking on the folder). You will get a compressed file named quiz04.zip. (Change folder view option in Control Panel if you do not see the file extensions.)
4. Submit quiz04.zip via D2L.

Grading Policy:

- Follow each instruction above very carefully and precisely. Each violation is subject to deduction. These are some of possible violations, but is not limited to:
  - class and file names (including case sensitivity)
  - use of 'package'

- use of strict commands and queries
- input validation
- submission: a compressed folder (not individual files).