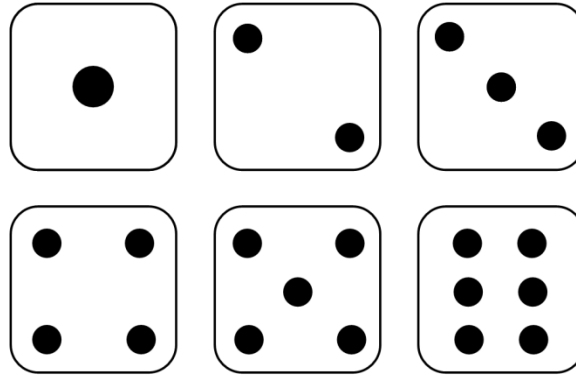# Programming Assignment 5 - Honors

Created by Tyler Greer

Objectives:
- Practice loops
- Practice using random numbers
- **NO ARRAYS!**
- **DON'T USE THE PARSE METHODS!**

Assignment:
For this assignment, you are going to recreate a dice game called "Run for it!". The goal of the game is to roll numbers that can be arranged into sequences of numbers that start with one (ex: 1, 2 or 1, 2, 3, etc.). This is a two player game. The players alternate turns, starting with player 1. During their turn, the player rolls 6 dice at once. If no sequences can be made from the numbers rolled, the player gets 0 points. If they do roll one or more sequences, for each number in the sequence(s), they receive 5 points and get to roll the dice again. The player gets to continue to roll the dice until the roll has no sequences. The first player that reaches 100 points wins the game.

The program will first display an introduction and the rules of the game. It asks player one to enter "R" to roll all 6 dice. The roll is displayed. Then, if any sequences are found, those are displayed separately, and points gained from them are added to player 1's score. If no sequences are found, it says "No sequences!" and no points are added. This process repeats for player 2. The players alternate turns until one player reaches 100. At the end, the winning player is declared, and then the program ends.

Game Notes:
- All sequences must start with one.
- No numbers can be skipped in the sequence.
- The smallest sequence possible is 1, 2.
- If a player rolls a sequence, they get to roll the dice again until they don't get any sequences.

Programming Notes:
- The dice rolls are generated using the Random class.
- Dice roll **MUST** be stored as a string.
- After a number in a sequence is found, it should be removed from the dice roll string.
- substring and/or chartAt methods will be useful.
- You don't need to input validation.

Sample Execution:

```
Run For It! - A Dice Game

Rules
=====
1. 2 players take turns rolling 6 dice.

2. If roll contains a sequence,
   same player gets another turn.

3. If roll doesn't contain a sequence,
   next player's turn.

4. Get points if roll contains one or
   more sequences.

     1. Sequence must start from 1 and
        not skip any numbers.

     2. Get 5 points for each number
        in the sequence(s).

5. Get no points if roll contains
   no sequences.

6. First player to get to 100 points
   wins.

======================================

Player 1 - Enter R to roll: R

Roll: 3 1 4 5 5 5

No sequence!

Player 1 Score: 0
Player 2 Score: 0

======================================

Player 2 - Enter R to roll: R

Roll: 4 1 5 4 2 3

Sequence 1: 12345

Player 1 Score: 0
Player 2 Score: 25

--------------------------------------

Player 2 - Enter R to roll: R

Roll: 4 1 3 2 5 1
```

```
Sequence 1: 12345

Player 1 Score: 0
Player 2 Score: 50

--------------------------------------

Player 2 - Enter R to roll: R

Roll: 6 2 4 4 2 4

No sequence!

Player 1 Score: 0
Player 2 Score: 50

======================================

Player 1 - Enter R to roll: R

Roll: 2 4 1 2 6 5

Sequence 1: 12

Player 1 Score: 10
Player 2 Score: 50

--------------------------------------

Player 1 - Enter R to roll: R

Roll: 2 1 5 4 6 2

Sequence 1: 12

Player 1 Score: 20
Player 2 Score: 50

--------------------------------------

Player 1 - Enter R to roll: R

Roll: 6 4 5 2 5 2

No sequence!

Player 1 Score: 20
Player 2 Score: 50

======================================

Player 2 - Enter R to roll: R

Roll: 2 6 6 2 5 4
```

No sequence!

Player 1 Score: 20
Player 2 Score: 50

==

Player 1 - Enter R to roll: R

Roll: 5 1 1 4 1 4

No sequence!

Player 1 Score: 20
Player 2 Score: 50

==

Player 2 - Enter R to roll: R

Roll: 3 2 1 2 2 6

Sequence 1: 123

Player 1 Score: 20
Player 2 Score: 65

Player 2 - Enter R to roll: R

Roll: 4 5 2 1 5 6

Sequence 1: 12

Player 1 Score: 20
Player 2 Score: 75

Player 2 - Enter R to roll: R

Roll: 6 5 3 6 3 4

No sequence!

Player 1 Score: 20
Player 2 Score: 75

==

Player 1 - Enter R to roll: R

Roll: 1 6 5 1 4 5

No sequence!

Player 1 Score: 20
Player 2 Score: 75

==

Player 2 - Enter R to roll: R

Roll: 2 6 2 5 1 3

Sequence 1: 123

Player 1 Score: 20
Player 2 Score: 90

Player 2 - Enter R to roll: R

Roll: 2 5 2 2 4 6

No sequence!

Player 1 Score: 20
Player 2 Score: 90

==

Player 1 - Enter R to roll: R

Roll: 6 3 1 6 6 1

No sequence!

Player 1 Score: 20
Player 2 Score: 90

==

Player 2 - Enter R to roll: R

Roll: 1 4 4 3 6 2

Sequence 1: 1234

Player 1 Score: 20
Player 2 Score: 110

==

                 Player 2 Wins!

==

Requirements:
- Use an updated comment block
- Your program should use the following comment block at the very beginning of your program.

  ```
  // Name:  Your Name                        Date Assigned: Fill in
  //
  // Course:  CSCI 2003 60357               Date Due:  Fill in
  //
  // Instructor: Ms. Greer
  //
  // File name:  Fill in
  //
  // Program Description:  Brief description of what the program does.
  ```

- Use appropriate comments throughout the program
- Make good use of whitespace
- Your output should look exactly like the sample output if using the same data.

Deliverables:
- DiceGame.java file
- Upload 1 file to Moodle

Grading:

Total Points	15 points
DiceGame class	**15 points**
Display rules.	0.5 point
Starts with player 1.	0.5 point
Uses a while loop to run the game until one player reaches 100 or more.	1 point
Asks user to enter "R" to roll.	0.5 point
Simulates rolling 6 dice using Random class with a loop.	1 point
Stores numbers from roll in a string.	1 point
Displays the 6 dice rolled.	0.5 point
Determines if there is at least one sequence.	1 point
If there is at least one sequence:	
Display each sequence.	0.5 point
Each sequence is numbered (Sequence 1, Sequence 2, etc.).	1 point
Update correct players score.	1 point
Score is calculated correctly.	1 point
Player gets another turn.	1 point
When asked to enter "R" to roll, it shows the correct player.	1 point
If there are NO sequences, display nothing.	0.5 point
Display player 1 and player 2's current score after last roll.	1 point
Change to next players turn.	1 point
Display winning player.	1 point
Not enough comments/whitespace	**-1 point**
Output does not match the sample executions given in the assignment	**-1 point**
Bad variable names, method names, and/or class names	**-1 point**