Assignment 01

CSIS 2175 – Section 004

The assignment contributes 5% to the overall grades. Students are required to submit the assignment to blackboard not later than **Feb 01, 2021 15:30 PST**. **NO LATE SUBMISSION** will be accepted. You may submit your work multiple times, but only the last submission will be graded.

Assignment could undergo a similarity check. If plagiarism is detected or students are found to have copied from each other, marks would be deducted.

Grading:

- Correctness of the program: 80%
- Programming style/comments/clarity: 10%
- Correctness of output format: 10%

Problem 1:

Description:

You are asked to write a Java program about a part of the award-winning game called *Qwirkle*. The description of Qwirkle can be found in https://en.wikipedia.org/wiki/Qwirkle.

Students are required to submit two java files, QwirklePlay.java and Qwirkle.java

Marks Allocated 10

In this assignment,

1. Create a class 'Qwirkle' with two attributes: colour and shape.
2. The digits 1-6 are used to represent the colours of the tile (Red, Green, Blue, Yellow, Orange, Pink).

3. The digits 1-4 are used to represent the shape of the tile (Square, rectangle, oval, triangle).
4. Each object of the class Qwirkle represents one tile to be used in the game.
5. Define a constructor to create object of class Qwirkle with random shape and colour.
6. Define a method to compare two tiles for equality. If a randomly generated Qwirkle object has shape and colour same as an existing tile, generate a new tile.
   Qwirkle object 1 has colour=green and shape=rectangle
   Qwirkle object 2 has colour=green and shape=rectangle
   Then generate new random shape and colour for object 2.

7. Save the class as Qwirkle.java
8. Create a main class QwirklePlay.java.
9. Player 1 picks 3 tiles with random values for colour and shapes.
10. Player 2 picks 3 tiles with random values for colour and shapes.
11. Tiles of player 1 should be unique.
12. Tiles of player 2 should be unique.
13. For player 1 Increment score equal to same-coloured (but unique) tiles.
14. For player 1Increment score equal to same-shaped (but unique) tiles.
15. For player 2 Increment score equal to same-coloured (but unique) tiles.
16. For player 2 Increment score equal to same-shaped (but unique) tiles
17. Display colour and shape of all the tiles of Player 1.
18. Display colour and shape of all the tiles of Player 2.
19. Player with higher score wins. Display Tie if score of both the players is same.

Some sample outputs are:

```
Player 1
==================================
Shape   Colour
2       4
3       6
4       5


Player 2
==================================
Shape   Colour
1       4
4       3
4       4
Player 2 won with score 2
```

```
Player 1
==================================
Shape   Colour
3       2
3       4
4       2


Player 2
==================================
Shape   Colour
2       1
2       6
1       6
Tie with score 2
```

```
Player 1
========================================
Shape   Colour
1       1
1       6
1       2


Player 2
========================================
Shape   Colour
2       1
2       4
1       5
Player 1 won with score 2
```

Bonus if output is displayed as

```
Player 1
================================================
Colour  Shape
================================================

Orange  Triangle
Green   Rectangle
Red     Square


Player 2
================================================
Colour  Shape
================================================

Yellow  Oval
Yellow  Rectangle
Green   Oval


Player 2 won with score 2
```

```
Player 1
================================================
Colour  Shape
================================================

Orange  Triangle
Red     Triangle
Orange  Rectangle


Player 2
================================================
Colour  Shape
================================================

Blue    Oval
Yellow  Oval
Green   Triangle


Player 1 won with score 2
```

```
Player 1
========================================
Colour  Shape
========================================

Pink    Triangle
Pink    Oval
Yellow  Rectangle


Player 2
========================================
Colour  Shape
========================================

Blue    Square
Blue    Rectangle
Orange  Oval


Tie with score 1
```

Problem 2

Write a program MyDice.java that takes an integer N, and rolls 5 fair six-sided dice, N times. Display the output of rolling dice. Calculate how many times you received at least one 6.

Students are required to submit MyDice.java.

Marks Allocated 5

Sample output

```
How many times to roll the dice
10
4 2 4 5 6
4 4 3 5 5
1 1 6 6 2
6 2 2 5 4
1 4 6 3 2
6 1 1 5 4
2 3 6 2 1
4 5 5 5 1
5 6 4 6 2
2 6 2 2 2
6 appeared at least once 8 times
```

```
How many times to roll the dice
5
5 6 2 6 4
4 2 4 3 4
4 3 2 6 4
1 5 3 6 2
1 3 2 5 2
6 appeared at least once 3 times
```

To randomly generate a number

```
(int)(Math.random() * Maximum + Minimum)
```

In order to see the sample executions QQwirkle.jar and TenDiceAssign.jar files have been uploaded in the Blackboard. In order to execute,

1. Download QQwirkle.jar.
2. Open command prompt
3. Move to the folder in which jar file has been stored (by using Cd command)
4. Type 'java -jar QQwirkle.jar'