Assignment 2 – Spring 2020

Skee-Ball is a popular classic redemption arcade game. The aim of the game is to roll a ball up an inclined lane and over a hump that jumps the ball into one of the bullseye rings. The objective of the game is to collect as many points as possible by having the ball fall into holes in the rings which have progressively increasing point values. The full description is available on Wikipedia (https://en.wikipedia.org/wiki/Skee-Ball)

- ✓ ***Please do your own work, sharing and/or copying code and/or solution ideas with/from others will result in a grade of 0 and disciplinary actions for all involved parties. If you run into any problems and have done your best to solve them, please see me before/after class or e-mail me.***
- ✓ ***If your submission is late, you will incur a 20% deduction of your score for each late day.***

In this assignment, we will write a Java program that simulates the game Skee-Ball. The assignment consists of a single Java class as described next. You are required to:

1. Properly comment your code. Comments should precede variables, methods, and major steps in your code.
2. Must use each of the following at least once in your solution (1) a WHILE (or a DO…WHILE) loop, (2) a FOR loop; (3) and a switch.

Class minimum members:

You may add additional members. However, your class should contain at least the following:

- Class contains one private STATIC and FINAL member called MAX_PLAYS. Initialize this member to 8.
- $main(\ )$. Performs the following
  - Prompt the user to enter the number of plays. This number must be between 1 and MAX_PLAYS (inclusive). If a number outside this range is entered, show an error message and prompt the user again. See Figure 1.
  - Once a valid number is entered, invoke the method $play(\ )$. Next, invoke the method $showStats(\ )$. Note that these methods are NOT static and cannot be invoked directly. Therefore, they must be invoked through an instance of your class.
  - After the method $showStats(\ )$ finishes, the game ends and the program terminates.
- $play(\ )$:
  - Must NOT be made STATIC. You may decide on any parameters the method may receive.
  - Using a random number generator, simulate a roll of the ball. Based on the number generated, assign the paly to a specific hall (points) using the percentages shown in Table 1.

    Hints:
    1. Create a separate method to return the hall value. This makes the code more manageable.
    2. Percentages can be simulated using the random number generated. For example, if the random number $(\mathcal{X})$ is between 1 and 10 $(1 \leq \mathcal{X} \leq 10)$, then a 50% is achieved if $1 \leq \mathcal{X} \leq 5$, a 20% is achieved if $6 \leq \mathcal{X} \leq 7$ …
  - The method repeats the play simulation a number of times equal to the value entered by the user in $main(\ )$
  - With every play, print a message that indicates the play number and the points assigned to that play (Figure 2).
- $showStats(\ )$:
  - Must NOT be made STATIC. You may decide on any parameters the method may receive.
  - Displays the results of the user's simulated plays along with the total points. See Figure 2
  - Note that this method's output is different from the output of $play(\ )$. This is simply a summary of all the simulated plays.

- Hint: $printf(\ \ )$ may come in handy here.

Hall value	Percentage of the time
80 points	5%
60 points	10%
40 points	15%
20 points	15%
10 points	20%
0 points	35%

Table 1: Hall values and percentages

Grading:

Item	Points
Comments	10
Static member	5
WHILE or DO…WHILE loop	5
FOR loop	5
Switch	5
$main(\ \ )$: ✓ Prompt for a number between 1 and 8 ✓ Invalid value check ✓ Proper termination	20
$play(\ \ )$ ✓ Simulation times based on the number entered (1-8) ✓ Assignment of points	20
$showStats(\ \ )$ ✓ Formatted output as shown in figures below	20
	100

Sample Outputs:

```
How many plays (1-8)? 9
Invalid input. Please enter a number between 1 and 8.
How many plays (1-8)? -1
Invalid input. Please enter a number between 1 and 8.
How many plays (1-8)? 0
Invalid input. Please enter a number between 1 and 8.
How many plays (1-8)?
```

Figure 1: Out-of-range entry. User is prompted again

```
How many plays (1-8)? 5
Rolling ball #1. Landed in 80
Rolling ball #2. Landed in 10
Rolling ball #3. Landed in 20
Rolling ball #4. Landed in 20
Rolling ball #5. Landed in 10

+-------+-------+
  Play #    Score
+-------+-------+
     1     80
     2     10
     3     20
     4     20
     5     10

Total: 140
```

```
How many plays (1-8)? 8
Rolling ball #1. Landed in 20
Rolling ball #2. Landed in 0
Rolling ball #3. Landed in 40
Rolling ball #4. Landed in 0
Rolling ball #5. Landed in 20
Rolling ball #6. Landed in 0
Rolling ball #7. Landed in 0
Rolling ball #8. Landed in 0

+-------+-------+
  Play #    Score
+-------+-------+
     1     20
     2      0
     3     40
     4      0
     5     20
     6      0
     7      0
     8      0

Total: 80
```

Figure 2: Sample Runs