Assignment 2 – Spring 2021

Due Date: by Sunday March 14, 2021 11:59PM

How to submit: upload JAVA files to Blackboard

Please note:
- ✓ This is an individual assignment; please do your own work. Sharing and/or copying code in part or whole with/from others will result in a grade of 0 and disciplinary actions for all involved parties. If you run into problems and have done your best to solve them, please talk to me during office hours or by e-mail.
- ✓ There is a 25% grade deduction for every day the assignment is late unless prior permission is granted.

Preamble

Figure 1[i]

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)[i].

In this assignment, you will write a Java program that simulates the game of Skee-Ball. The assignment consists of a single Java class as described below. Additionally, you must use each of the following structures at least once in your code:

1. WHILE (or Do…While) loop
2. FOR loop
3. SWITCH structure
4. Static and final class member called MAX_PLAYS with a value of 8
5. Two *non-static* methods as described below. You may add additional *non-static* methods if you think your implementation can benefit from them. Your program should NOT contain any $static$ methods besides $main()$.

Game flow:

1. Prompt the user to enter the number of plays. Ensure that the number entered is between 1 and MAX_PLAYS (inclusive). If the number entered is outside of the two limits, display an error message and prompt the user again. See Figure 2.
2. After the user enters a valid input, invoke the method $simulatePlay()$
3. Finally, invoke the method $displayStats()$. The program should terminate afterwards.

Required Methods:

I. $main()$:
- ✓ The only static method in the program and contains the game's flow described earlier. Remember that static method can only invoke non-static ones through an instance of a class.

II. $simulatePlay()$:
- ✓ Non-static
- ✓ Add parameters if needed
- ✓ Using Java's random number generator class, generate a random number to simulate the throw of the ball. Assign the play a value based on the percentages shown in Table 1 (See the hint below the table.). You may, also, choose to create a separate method to return the hall value. This improves the code's manageability and readability.
- ✓ The method repeats the play simulation several times equal to the value entered by the user in $main()$

- ✓ After each play iteration, print a message which shows the play number and the points assigned (Figure 3).

III. $displayStats()$:

- ✓ Non-static
- ✓ Add parameters needed
- ✓ Displays the results of the simulated plays along with the total points. See Figure 3 and Figure 4
- ✓ Note that this method's output is different from that of $simulatePlay(\ )$. This is simply a summary of all the simulated plays.
- ✓ Hint: $printf()$ simplifies printing here.

Grading:

Item	Points
Comments (Javadoc and major steps)	10
Compiles and correct output	10
$Static\ member$	5
$WHILE$ or $DO \ldots WHILE$ loop	5
FOR loop	5
$Switch$	5
$main()$: ✓ Prompt for a number between 1 and 8 ✓ Invalid value check ✓ Proper termination of resources	20
$simulatePlay()$ ✓ Simulation times based on the number entered (1-8) ✓ Assignment of points	20
$showStats(\ )$ ✓ Formatted output as shown in Figure 3 and Figure 4	20
	100

Tables:

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

Percentages Hint:

One way to simulate percentages is to use the random number generator with a proper range divided into groups. For example, if the random number X is between 1 and 10 (i.e., $1 \leq X \leq 10$), then 50% maybe achieved when $1 \leq X \leq 5$, 20% when $6 \leq X \leq 7$ …

[I] https://commons.wikimedia.org/wiki/File:Skee_Ball.JPG#/media/File:Skee_Ball.JPG

Figures:

```
Enter the number of plays (1-8)? 0
Invalid input. Please enter a number between 1 and 8.
Enter the number of plays (1-8)? -1
Invalid input. Please enter a number between 1 and 8.
Enter the number of plays (1-8)? 100
Invalid input. Please enter a number between 1 and 8.
Enter the number of plays (1-8)?
```

Figure 2: Invalid entry message and re-prompt

```
Enter the number of plays (1-8)? 5
Rolling ball #1. Landed in 0
Rolling ball #2. Landed in 10
Rolling ball #3. Landed in 20
Rolling ball #4. Landed in 80
Rolling ball #5. Landed in 40

+-------+-------+
  Play #    Score
+-------+-------+
     1      0
     2     10
     3     20
     4     80
     5     40

Total: 150
```

Figure 3: Sample simulation using 5 plays (number will vary due to the random number generator)

```
Enter the number of plays (1-8)? 8
Rolling ball #1. Landed in 40
Rolling ball #2. Landed in 10
Rolling ball #3. Landed in 10
Rolling ball #4. Landed in 10
Rolling ball #5. Landed in 10
Rolling ball #6. Landed in 0
Rolling ball #7. Landed in 10
Rolling ball #8. Landed in 20

+-------+-------+
  Play #    Score
+-------+-------+
     1     40
     2     10
     3     10
     4     10
     5     10
     6      0
     7     10
     8     20

Total: 110
```

Figure 4: Sample simulation using 8 plays (number will vary due to the random number generator)