SpinnerGame FRQ

SHOW ALL YOUR WORK. REMEMBER THAT PROGRAM SEGMENTS ARE TO BE WRITTEN IN JAVA.

Assume that the classes listed in the Java Quick Reference have been imported where appropriate. Unless otherwise noted in the question, assume that parameters in method calls are not null and that methods are called only when their preconditions are satisfied.
In writing solutions for each question, you may use any of the accessible methods that are listed in classes defined in that question. Writing significant amounts of code that can be replaced by a call to one of these methods will not receive full credit.

This question involves a game that is played with multiple spinners. You will write two methods in the *SpinnerGame* class below.

```
public class SpinnerGame
{
   /** Precondition: min < max
    * Simulates a spin of a spinner by returning a random integer
    * between min and max, inclusive.
    */
   public int spin(int min, int max)
   { /* to be implemented in part (a) */ }

   /** Simulates one round of the game as described in part (b).
    */
   public void playRound()
   { /* to be implemented in part (b) */ }
}
```

(a) The *spin* method simulates a spin of a fair spinner. The method returns a random integer between *min* and *max*, inclusive. Complete the *spin* method below by assigning this random integer to *result*.

```
/** Precondition: min < max
 * Simulates a spin of a spinner by returning a random integer
 * between min and max, inclusive.
 */
public int spin(int min, int max)
```

SpinnerGame FRQ

In each round of the game, the player and the computer each spin a spinner. The player spins a spinner numbered 1 to 10, inclusive, whereas the computer spins a spinner numbered 2 to 8, inclusive.

Based on the results of the spins, a message is printed in the formats shown in the examples below.

If the player obtains a higher result than the computer, the player gains a number of points equal to the positive difference between the spins. If the computer obtains a higher result than the player, the player loses a number of points equal to the positive difference between the spins.

In the event of a tie, the player and the computer each spin the spinner a second time. If the sum of the player's two spins are greater than the sum of the computer's two spins, the player gains one point. If the sum of the computer's two spins are greater than the sum of the player's two spins, the player loses one point. In the event of a tie after two spins, the round is reported as a tie and the player's score does not change.

Examples of the `playRound` method's intended behavior are shown in the following table.

Player Spin #1	Computer Spin #1	Player Spin #2	Computer Spin #2	Printed String
9	6			You win! 3 points
3	7			You lose. -4 points
4	4	6	2	You win! 1 points
6	6	1	2	You lose. -1 points
1	1	8	8	Tie. 0 points

(b) Complete the playRound method below. You must use the spin method appropriately in order to earn full credit.

```
/** Simulates one round of the game as described in part (b).
 */
public void playRound()
```