Assignment 3: Blackjack Solitaire

This assignment focuses heavily on your usage of good design principles (DRY, principle of least surprise, coupling and cohesion). You will be implementing object-oriented design working from specifications, creating CRCs, and then finally building it out in Java.

This is also the first assignment in which we do not tell you all the classes and all the methods that you need to write. We encourage you to ask us questions via the discussion forums.

*The activities in **Week 3 Recitation** will help you practice for this assignment, so it is highly recommended that you attend the live sessions or review the recordings and be sure to attempt the recitation activities.*

Learning Goals

This assignment is designed to reinforce the following concepts:

- Object-oriented design (using CRCs)
- Learning how to test your assignment thoroughly
- Programming a game

End Product of the Assignment

You will implement the card game called Blackjack Solitaire:
http://www.solitairenetwork.com/solitaire/blackjack-square-solitaire-game.html

Familiarize yourself with the game by playing the online version before considering programming. Make sure you know how to score. Our goal is to enforce the rules, allow the user to play 1 round of the game and then score the game.

This week there is a Graded Discussion where you will practice playing the game and post your high scores.

The game is a solo game, so in that sense it is like Solitaire, but all of the scoring comes from Blackjack. In Blackjack, a hand's score should stay at or below a value of **21**. In this game, Blackjack hands are scored from nine hands formed by each of the four rows and five columns of the grid of cards laid out. To play the game, you draw cards one at a time from the deck and place them on the grid. Once placed, a card cannot be moved. The four discard spots allow one to ignore four cards by placing them in the discard spots rather than on the grid. Once all 16 spots in the grid have cards, a score is calculated.

How to Score Blackjack Solitaire

Every card has a value. If it is from 2 to 10, the value is the number associated with the card. If it is a face card, that is, if you have king, jack or queen, then it is worth 10 points.
Finally, the trickiest card to score is the ace, which counts for 11 or 1, depending upon which gives you a higher score without going to bust and going over 21.

Hand	Points	Explanation
Blackjack	10	Blackjack is two cards that total 21
21	7	3, 4, or 5 cards totaling 21
20	5	Hand totals 20
19	4	Hand totals 19
18	3	Hand totals 18
17	2	Hand totals 17
BUST	0	Hand totals 22 or more
16 and others	1	Hand totals 16 or less

Requirements

Once you have played the game for a bit and understand the way it works, here are the requirements.

1. Your game will play one hand and then exit. That is, the 16 spots on the grid will be filled, and at most 4 spots on the discard will be filled. Then a score is calculated; then the game will quit.

2. The game begins by shuffling the deck and dealing a card. The 16 grid spots and four discard spots are numbered 1-20 as shown below.

1	2	3	4	5
6	7	8	9	10
	11	12	13	
	14	15	16	

The discard spots are 17, 18, 19, and 20.

You will repeatedly prompt the user to enter a number indicating where to place the dealt card.

3. For displaying the state of the game at any point, we will use the convention that the card is going to be displayed with the rank combined with the suit. For example, the King of Hearts is KH. The 3 of diamonds is 3D. Each spot on the grid is either going to have a number, or is going to contain the string representation of the card.

```
1       KH      3       4       5
2D      7       8       9       AS
        11      12      13
        14      7C      16
```

4. After each card is placed, a new card is automatically dealt.

5. When the 16th card is placed in the grid, the game ends and the score is calculated.

6. Error checking:

- If the user enters the number of a spot that already has a card, you will print an error message and repeat the prompt.

7. You **have** to design the following classes:

- A Card class that represents a single card.
- A Deck class that represents a deck of cards.
- A BlackjackSolitaire class that represents the entire game

You are encouraged to use the CRC technique and make even more classes.

At the end, we want you to create the following class that actually runs the game:

```
public class BlackjackSolitaireRunner {
     public static void main(String[] args) {
          BlackjackSolitaire bjs = new BlackjackSolitaire();
          bjs.play();
     }
}
```

This file is supplied to you and can be used as provided.

Breaking Down the Specifications into Smaller Steps

In order to help you a bit more, let us break up the entire game into smaller segments:

(a) Display initial state of the game.

(b) Shuffle deck.

(c) Deal a card.

(d) Allow user to make a move. How does the user make a move? We prompt the user to supply a position where they want to put the card and then we take that information and move the dealt card to the appropriate position. When we allow the user to make a move, we also want to error check the move. **Hint:** do the error checking in a separate function/functions.

(e) Display current state of the game. (Display the board after each turn.)

(f) Repeat above three steps (deal card, place card, display game) over and over until the game is complete.

(g) At the point where all 16 points of the table are filled, print a message telling the user you are going to score the hands and then pass the state of the table to a scoring function.

(h) The final message to display is just the score of your table.

(i) Print a message saying the game is done.

What to Submit

Please submit at least these 4 files:
Card.java
Deck.java
BlackjackSolitaire.java
BlackjackSolitaireRunner.java

There will be other files that you will create (hint hint). Make sure that you submit them as well. They will count as part of your grade for the assignment. Remember that if your code does not compile, we will not be able to give you a score. Also those extra files might amount to your design being even better. (Remember, one of the learning goals for this assignment is design!)