File - C:\CSC120\Sulley_Lab8\src\Deck.java

```java
import java.time.chrono.MinguoChronology;
import java.util.ArrayList;
import java.util.Random;

/**
 * Simulates a deck of 52 playing cards.
 */
public class Deck {
    private static final int NUMBER_OF_CARDS=52;
    private static final int NUMBER_OF_SUITS=4;
    private static final int CARDS_IN_SUIT=13;

    private ArrayList<Card> theCards;
    private boolean shuffled;

    /**
     * Constructs a new ordered deck of playing cards
     */
    public Deck()
    {
        theCards = new ArrayList<Card>(NUMBER_OF_CARDS);
        shuffled=false;
        generate();
    }

    /**
     *generates all the Cards in a deck and adds them to the ArrayList of Cards
     */
    private void generate(){
        int MIN_RANK = 2;
        for(String suit : Card.SUITS){
            for(int i=MIN_RANK; i-MIN_RANK < CARDS_IN_SUIT ; i++ ){
                theCards.add(new Card(i, suit));
            }
        }
    }

    /**
     * Deals out next card in deck; returns null if no cards left
     *
     * @return next card in deck or null if deck empty
     */
    public Card deal() {

        if(this.isEmpty()) {
            return null;
        }
        else{
            Card nextCard;
```

```
50              if(shuffled == true){
51                  nextCard = theCards.get(new Random().
   nextInt( theCards.size()));
52              }
53              else{
54                  nextCard = theCards.get(theCards.size() -
   1);
55              }
56              theCards.remove(nextCard);
57              return nextCard;
58          }
59      }
60
61      /** determines if deck has any cards left in it
62       *
63       * @return true if Deck empty; else false
64       */
65      public boolean isEmpty(){
66          return theCards.isEmpty();  // erase this. It's
   just here so code will compile.
67      }
68
69      /**
70       * Shuffles the cards
71       */
72      public void shuffle()
73      {
74          shuffled = true;
75      }
76
77      /** Returns number of undealt cards left in the deck
78       *
79       * @return number of undealt cards in the deck
80       */
81      public int size()
82      {
83          return theCards.size();  // erase this. It's just
    here so code will compile.
84      }
85
86      /**
87       * Reset the deck by gathering up all dealt cards.
88       * Postcondition: Deck contains all cards and is NOT
   shuffled
89       */
90      public void gather() {
91          theCards.clear();
92          generate();
93          shuffled = false;
94      }
95
96      /**
```

```
 97      *  DEBUGGING METHOD: prints out stats of the given list of cards, that is, what was dealt.
 98      *  Prints the remaining number of cards of each suit and of each rank.
 99      *
100      *  @param cardList list of cards that are (were) in the deck
101      */
102     public void printStats(ArrayList<Card> cardList)
103     {
104         int Hcount=0;
105         int Dcount=0;
106         int Scount=0;
107         int Ccount=0;
108         int[] ranks = new int[CARDS_IN_SUIT];
109         int size=cardList.size();
110         for (int i=0; i<size; i++)
111         {
112             int val = cardList.get(i).getRank();
113             String suit = cardList.get(i).getSuit();
114             if (suit.equals("clubs"))
115                 Ccount++;
116             else if (suit.equals("diamonds"))
117             Dcount++;
118             else if (suit.equals("spades"))
119             Scount++;
120             else if (suit.equals("hearts"))
121                 Hcount++;
122             ranks[val-2]++;  // deck RANKS run from 2-14 so need to subtract 2
123         }
124         System.out.println("***PRINTING DECK STATS***");
125         System.out.println("# clubs: " + Ccount);
126         System.out.println("# diamonds: " + Dcount);
127         System.out.println("# hearts: " + Hcount);
128         System.out.println("# spades: " + Scount);
129 
130         System.out.print("Card:\t");
131         for (int j = 2; j< Card.RANKS.length; j++) {
132             System.out.print(Card.RANKS[j]+"\t");
133         }
134         System.out.println();
135         System.out.print("Qty:\t");
136         for (int j=0; j<ranks.length; j++) {
137             System.out.print(ranks[j] + "\t");
138             if (j>8) {  // indices 9-12 are Jack thru Ace
139                 System.out.print("\t");
140             }
141         }
142         System.out.println("\n");
143     }
144 
```

```
145     /**
146      *  DEBUGGING METHOD: prints out stats of this Deck
    object
147      *  Prints the remaining number of cards of each suit
     and of each rank.
148      */
149     public void printStats() {
150         printStats(theCards);
151     }
152 
153 
154 }
155 
```