Name:   
Date:

**17.06 Elevens Lab Worksheet**

**Directions**: Make note of your responses to the following questions as you work through the activities and exercise in the lesson.

**Activity 1 Exercise Results**

1. From step a, paste your Card class constructor below.

public Card(String cardRank, String cardSuit, int cardPointValue)

{

rank = cardRank;

suit = cardSuit;

pointValue = cardPointValue;

}  
1a. From step c, paste your matches method below.

public boolean matches(Card otherCard)

{

if(this.rank().equals(otherCard.rank()) && this.suit().equals(otherCard.suit) && this.pointValue() == otherCard.pointValue()){

return true;

}

else{

return false;

}

}

1. Paste the results of running the CardTester.java class below.

\*\*\*\* Tests Card 1: ace of hearts \*\*\*\*

rank: ace

suit: hearts

pointValue: 1

toString: hearts of ace (point value = 1)

\*\*\*\* Tests Card 2: king of clubs \*\*\*\*

rank: king

suit: clubs

pointValue: 5

toString: clubs of king (point value = 13)

\*\*\*\* Tests Card 3: jack of hearts \*\*\*\*

rank: jack

suit: hearts

pointValue: 3

toString: hearts of jack (point value = 11)

\*\*\*\* Tests matching method between jackHearts1 and jackHearts2 \*\*\*\*

matching: true

\*\*\*\* Tests matching method between jackHearts1 and kingClubs \*\*\*\*

not matching: false

**Activity 2 Exercise Results**

1. From step a, paste your Deck class constructor below.

cards = new ArrayList<Card>();

for(int j = 0, k = 0; j<ranks.length && k<values.length; j++, k++){

for(int i = 0; i<suits.length; i++){

String rankJ = ranks[j];

String suitsI = suits[i];

int valuesK = values[k];

cards.add(new Card(rankJ, suitsI, valuesK));

}

}

size = cards.size();  
  
From step b, paste your isEmpty method below.

if(size == 0)

{

return true;

}

else

{

return false;

}  
  
From step d, paste your deal method below.

if (size == 0)

{

return null;

}

else

{

size--;

return cards.get(size);

}  
  
Paste the results of running the DeckTester.java class below.  
\*\*\*\* Original Deck Methods \*\*\*\*

toString:

size = 15

Undealt cards:

8 of spades (point value = 8), 8 of clubs (point value = 8),

8 of hearts (point value = 8), 9 of spades (point value = 9),

9 of clubs (point value = 9), 9 of hearts (point value = 9),

jack of spades (point value = 11), jack of clubs (point value = 11),

jack of hearts (point value = 11), king of spades (point value = 13),

king of clubs (point value = 13), king of hearts (point value = 13),

queen of spades (point value = 12), queen of clubs (point value = 12),

queen of hearts (point value = 12)

Dealt cards:

isEmpty: false

size: 15

\*\*\*\* Deal a Card \*\*\*\*

deal: 8 of spades (point value = 8)

\*\*\*\* Deck Methods After 1 Card Dealt \*\*\*\*

toString:

isEmpty:

size:

\*\*\*\* Deal Remaining 5 Cards \*\*\*\*

\*\*\*\* Deck Methods After All Cards Dealt \*\*\*\*

toString:

isEmpty:

size:

\*\*\*\* Deal a Card From Empty Deck \*\*\*\*

deal:

## **Activity 2 Questions:**

1. Explain in your own words the relationship between a deck and a card.

A card is the object with rank, suit, and value while a deck contains all the cards.

1. Consider the deck initialized with the statements below. How many cards does the deck contain? \_6\_

String[] ranks = {"jack", "queen", "king"};

String[] suits = {"blue", "red"}; int[] pointValues = {11, 12, 13};

Deck d = new Deck(ranks, suits, pointValues);

1. Many card games are played with a deck of 52 cards. It is common for ranks to run from ace (highest) down to 2 (lowest). Suits are spades, hearts, diamonds, and clubs. A face card has point value 10; an ace has point value 11; point values for 2, …, 10 are 2, …, 10, respectively. Write the statements to declare and initialize the contents of the ranks, suits, and pointValues arrays so that the following statement initializes a deck as described.

Deck d = new Deck(ranks, suits, pointValues);

String [ ] ranks = {“2”, “3”, “4”, “5”, “6”, “7”, “8”, “9”, “10”, “jack”, “queen”, “king”, “ace”};

String [ ] suits = {“spades”, “hearts”, “diamonds”, “clubs”};

int [ ] pointValues = {2, 3, 4, 5, 6, 7, 8, 9, 10, 10, 10, 10, 11};

1. Does the order of elements of the ranks, suits, and pointValues arrays matter?

ranks and point values should be ordered in the same way. the suits can have a different order.