Name: Paul Chafetz  
Date: 5/14/2019

**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)

{

suit = cardSuit;

rank = cardRank;

pointValue = cardPointValue;

}

1. From step c, paste your matches method below.  
   public boolean matches(Card otherCard)

{

if (this.suit().equals(otherCard.suit()) && this.rank().equals(otherCard.rank()) && 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: ace of hearts (point value = 1)

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

rank: 5

suit: clubs

pointValue: 5

toString: 5 of clubs (point value = 5)

\*\*\*\* Tests Card 3: seven of spades \*\*\*\*

rank: 7

suit: spades

pointValue: 7

toString: 7 of spades (point value = 7)

\*\*\*\* Tests matches method \*\*\*\*

ace of hearts matches ace of hearts?: true

seven of spades matches ace of hearts?: false

five of clubs matches seven of spades?: false

**Activity 2 Exercise Results**

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

public Deck(String[] ranks, String[] suits, int[] values)

{

cards = new ArrayList<Card>();

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

Card c = new Card(ranks[i], suits[i], values[i]);

cards.add(c);

}

size = cards.size();

shuffle();

1. From step b, paste your isEmpty method below.

public boolean isEmpty()

{

if (size() == 0)

return true;

else

return false;

}

1. From step d, paste your deal method below.

public Card deal()

{

if (size() > 0) {

size -= 1;

return cards.get(size);

}

else

return null;

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

\*\*\*\* Original Deck Methods \*\*\*\*

toString:

size = 10

Undealt cards:

Jack of Spades (point value = 11), Ten of Hearts (point value = 10),

Eight of Diamonds (point value = 8), Ace of Diamonds (point value = 1),

Four of Hearts (point value = 4), King of Clubs (point value = 13),

Queen of Clubs (point value = 12), Three of Spades (point value = 3),

Jack of Diamonds (point value = 11), Five of Hearts (point value = 5)

Dealt cards:

isEmpty: false

size: 10

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

deal: Jack of Spades (point value = 11)

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

toString:

size = 9

Undealt cards:

Ten of Hearts (point value = 10), Eight of Diamonds (point value = 8),

Ace of Diamonds (point value = 1), Four of Hearts (point value = 4),

King of Clubs (point value = 13), Queen of Clubs (point value = 12),

Three of Spades (point value = 3), Jack of Diamonds (point value = 11),

Five of Hearts (point value = 5)

Dealt cards:

Jack of Spades (point value = 11)

isEmpty: false

size: 9

\*\*\*\* Deal Remaining 9 Cards \*\*\*\*

deal: Ten of Hearts (point value = 10)

deal: Eight of Diamonds (point value = 8)

deal: Ace of Diamonds (point value = 1)

deal: Four of Hearts (point value = 4)

deal: King of Clubs (point value = 13)

deal: Queen of Clubs (point value = 12)

deal: Three of Spades (point value = 3)

deal: Jack of Diamonds (point value = 11)

deal: Five of Hearts (point value = 5)

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

toString:

size = 0

Undealt cards:

Dealt cards:

Jack of Spades (point value = 11), Ten of Hearts (point value = 10),

Eight of Diamonds (point value = 8), Ace of Diamonds (point value = 1),

Four of Hearts (point value = 4), King of Clubs (point value = 13),

Queen of Clubs (point value = 12), Three of Spades (point value = 3),

Jack of Diamonds (point value = 11), Five of Hearts (point value = 5)

isEmpty: true

size: 0

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

deal: null

## **Activity 2 Questions:**

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

A card is a single card with its own unique suit, rank, and point value. A deck is a group (list) of cards together. A deck ArrayList is comprised of Card objects.

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

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

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

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

There are 3 cards in this deck.

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.

//note that there are 4 of each rank, 13 of each suit, and 4 of each point value 0-9, 16 tens, and 4 elevens

//I don’t want to write 3 arrays each of 52 possibilities because it would be very ugly and time-consuming, so I only show the first four elements of each array here

Deck d = new Deck(ranks, suits, pointValues);  
String[] ranks = {“one”, “one”, “one”, “one”, …, };

String[] suits = {“hearts”, “hearts”, “hearts”, “hearts”, …, };

String[] pointValues = {1, 1, 1, 1, …, };

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

Yes. This is because each index in the array is paired with its respective arrays to make up one card. Therefore, it is important that each card is correctly paired with its ranks[i], suits[i], and pointValues[i] in the for-loop.