**Activity 2 and Activity 3 Questions**

**Activity 2**

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

A deck is a Collection of Cards.

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);

6

1. The game of Twenty-One is played with a deck of 52 cards. Ranks run from ace (highest) down to 2 (lowest). Suits are spades, hearts, diamonds, and clubs as in many other games. A face card has point value 10; an ace has point value 11; point values for 2, ..., 10 are 2, ..., 10, respectively. Specify the contents of the ranks, suits, and pointValues arrays so that the following statement initializes a deck for a Twenty-One game.

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

ranks = {"Ace", "2", "3", "4", "5", "6", "7", "8", "9", "10", "Jack", "Queen", "King"}

suits = {"Clubs", "Hearts", "Diamonds", "Spades"}

pointValues ={11,2,3,4,5,6,7,8,9,10,10,10,10}

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

It is the order which they are initialized.

**Activity 3**

1. Write a static method named flip that simulates a flip of a weighted coin by returning either "heads" or "tails" each time it is called. The coin is twice as likely to turn up heads as tails. Thus, flip should return "heads" about twice as often as it returns "tails."  public static String flip() {
   * 1. int rand = (int)(Math.random()\*3);
        1. return rand < 2 ? “heads” : “tails”;
        2. }
2. Write a static method named arePermutations that, given two int arrays of the same length but with no duplicate elements, returns true if one array is a permutation of the other (i.e., the arrays differ only in how their contents are arranged). Otherwise, it should return false.

public static boolean arePermutations(int [] a; int [] b) {

for (int i : a) {

boolean found = false;

for (int j = 0; j < b.length; j++)

if (b[j] = i)

found = true;

if (!found)

return false;

}

return true;

}

1. Suppose that the initial contents of the values array in Shuffler.java are {1, 2, 3,  4}. For what sequence of random integers would the efficient selection shuffle change values to contain {4, 3, 2, 1}?

0,1,1,0