Elevens Lab

Activity 2:

1. A card is a subclass of the Deck class, meaning multiple cards are contained in a deck.
2. 6
3. String[] ranks = {”two”,”three”,”four”,”five”,”six”,”seven”,”eight”,”nine”, “ten”,"jack", "queen", "king",”ace”};

String[] suits = {"Diamonds", "Hearts",”Spades”,”Clubs”};

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

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

1. The order of ranks and pointValues matters because the ranks will each be assigned a respective point value.

Activity 3:

1. Public String flip()

{

Int newNum = random.nextInt(4);

If(newNum == 1 || newNum == 2)

{

return “heads”;

}

Else

{

Return “tails”;

}

1. public Boolean arePermutations(values1, values2)

{

Int trueCount = 0;

For (int i = 0; i<values1.length(); i++)

{

Int J = 0;

If (values1[i] == values2[j])

{

trueCount ++;

}

Else

{

If (j <= values2.length)

{

j++;

}

}

}

If (trueCount == values1.length)

{

Return true;

}

Else

{

Return false;

}

}

Activity 5:

1. The error is most likely in the isEmpty method, the problem is probably that the method is returning true no matter what the size of the deck is.
2. There’s an error in getSize(), probably because the for loop to get size is not implemented correctly
3. There’s an error in the shuffle method because the method shuffles the deck enough times that they return to their original states.
4. There’s an error in the deal() method which prevents the first card from being placed.

Activity 6:

1. 5 and 6,
2. Yes, there is no other way to remove a triplet
3. I believe it odes because when playing elevens you constantly have to account for which cards are left in the deck and which cards you should avoid eliminating in order to make future matches.

Activity 7:

1.

* cardsLeftInDeck
* totalCards
* deck

Activity 8;