Activity one questions:

n/a

Activity 2 questions:

1) A deck HAS (a) card(s).

2) The deck contains three cards because the values list has 3 items in it.

3) In the ranks array ["Ace", "King", "Queen", "Jack", "10", "9", "8", "7", "6", "5", "4", "3", "2"]

In the values array [11, 10, 10, 10, 10, 9, 8, 7, 6, 5, 4, 3, 2]

In the suits array ["Diamonds", "Hearts", "Spades", "Clubs"]

4) Yes the order matters because the order of ranks and the order of values correspond and that correspondance

is used to determine which card has which value.

Activity 3 questions:

1) public static String flip()

{

Random rand = new Random();

int r = rand.nextInt(3);

String h = "Heads";

String t = "Tails";

if( (r == 1) || (r ==2))

{

return h;

}

else

{

return t;

}

}

2) public static boolean arePermutations(int[]a1, int[]a2)

{

for(int i = 0; i < a1.length; i ++)

{

if(a1[i] == a2[i])

{

return false;

}

}

return true;

}

3) 1, 2 , 3, 4

Activity 5:

1. User may have set the Boolean statement to return when size != 0 instead of size ==0
2. User may have accidentally removed the card by decrementing size by 2
3. User most likely copied and pasted variable data and forgot to modify it
4. User may have forgotten to create a deck with cards first, or to deal a card
5. Received a bounds error, meaning that the user most likely tried to iterate incorrectly through a for loop.

Activity 6:

1. 5spade and 6 club/ 6 club and 6 club
2. To win they must be king queen and jack, but otherwise if it isn’t king queen jack the player cant win
3. From my experience with the game there is no way to strategically win.. only luck. When I tried matching two cards of the same suit(equaling eleven) it rendered the same random replacement as matching two cards of dissimilar suits(equaling eleven).

Activity 7:

1. An array of randomly ordered cards in a deck An array that captures the last move /An integer to record how many cards are left in the deck
2. Shuffle the deck

Deal the first 9 cards

For (I = 0 < deck .length; I ++)

{

If (card1 + card2 == 11)

{

Deal 2 new cards

}

Else if( card1 , card2, card3 == jack, queen, king )

{

Deal 3 new cards

}

Else if ( no playable moves)

{

Print “You lost game over.”

}

Else()

{

beep sound

}

}

If (deck.length == 0)

{

Print “Congratulartions, you’ve won!”

}

}

1. Not entirely, some things still need to be implemented
2. A) In the constructor of the board and in the creation of a new game.

B) Within the isLegal method and within the replaceSelectedCards method

C) RETURNED: Jack(Hearts), 6( clubs), 2(Spades), Ace(Spades), 4(Hearts)

D) String str = “”;

For( Card i: cIndexes)

{

Str += I + “ “;

}

System.out.println(str);

E) The replaceSelectedCards method needs to call the cardIndexes method first because to remove cards you must first know their index values.

Activity 8:

1. The card games 10s, 11s, and 13s all have the same basic concept to add the cards up to a preselected value to continue to the next turn. And as I had found out earlier, the elevens game is completely based on luck so by default I must assume that the other games are built on luck and chance as well. Some differences include different board sizes and values for face cards.
2. Because the Board class is an abstract class the Board instance variables must get initialized with the ElevensBoard values is that the constructor for the a Board object must take in values from the ElevensBoard class as parameters.
3. isLegal and anotherPlayIsPossible. Yes, because they are able to override the parent class methods and able to make them their own.

Activity 9:

1. Because size could be an instance variable.
2. Because logically it does not make sense to create the methods in that class.
3. This design would not work, many key components are missing, and it is not very straight forward, and may require reworking other areas of the project.