Quentin Verkler
Mr. Ettlin
APCSA, Period 3
February 25, 2019

Elevens Questions

Activity 2

1.) A deck is made up of cards.
2.) The deck has 6 cards.
3.) Ranks are Ace to 2. Suits are spade, hearts, diamonds and clubs. pointValues are [11, 10, 10, 10, 10, 9, 8, 7, 6, 5, 4, 3, 2].
4.) Yes because ranks have assigned pointValues, so if you mess them up, ranks will have the wrong pointValues.

Activity 3

1.)
```
static String flip(){
        int r = (int)(Math.random() * 3);
        if(r == 0 || r == 1)
                return “heads”;
        else
                return “tails”;
}
```
2.)
```
static boolean arePermutations(int[] first, int[] second){
        boolean check = true;
        for(int i = 0; i < first.length; i++){
                for(int j = 0; j < second.length + 1; j++){
                        if(j == second.length + 1)
                                return false;
                        if(first[i] = second[j]){
                                i += 1;
                                j = 0;
                                }

                }
        }
        return check;
}
```
3.) r = 4, r = 3, r = 4

Activity 6

1.) 5 of spades and 6 of clubs, 6 of clubs and 5 of clubs
2.) The must be J, Q, and K because all other combinations would have already been taken.
3.) There is no strategy. It’s all up to chance, and the chance doesn’t change based on which card you choose.

Activity 7

1.) You would need card identities, suits, and values.
2.) Check if board is full

```
Check if board is full
        If not, then see if there are still cards in deck
                If yes, then fill board
        Else
                Don't fill board
If player selects cards that are valid pairs
        Remove and replace
If no more combos exist
        Game over
If board is empty
        Game won
```

3.) It has neither the card nor the deck class. It doesn't have something for game over or to check if pair is valid.
4.)
    a.) dealMyCards() is called at the end of the constructor.
    b.) The method isLegal() and anotherPlayIsPossible() should use the two methods
    c.) J of hearts, 6 of clubs, 2 of spades, A of spades, 4 of hearts
    d.) public static void printCards(ElevensBoard board){

```
public static void printCards(ElevensBoard board){
        List<Integer> cIndexes = bard.cardIndexes();
        for(int i = 0; i < cIndexes.size(); i++){
                if(cIndexes.get(i) != null)
                        System.out.println(cIndexes.get(i).toString());

        }
```

    e.) anotherPlayIsPossible() is necessary because if a play isn't possible, then there's no point calling the other methods.

Activity 8

1.) All use the same cards, deck, shuffle and deal methods. However, each has different isLega() definitions and anotherPlayIsPossible().
2.) You still need a ElevensBoard constructor. Give it no parameters and use super.
3.) The abstract methods are isLegal(List<Integer> selected cards) and anotherPlayIsPossible(). They cover all the differences because each can take in the different rules for each game but still say whether or not an action is legal or another pair exists.

Activity 9

1.) The way size works is the same for each game, so there isn't a need to make it abstract.
2.) The selection of cards being removed/replaced remains the same per game, so you can just put them into the Board class.

3.) No, you could not use isLegal and anotherPlayIsPossible with implementation polymorphically because you can only implement interfaces.