Activity 2

1. Deck has an aggregation relationship with Card because Deck has a card and card is owned by deck.
2. 6 Cards
3. Ranks = {“ace”, “2”, “3”, “4”, “5”, “6”, “7”, “8”, “9”, “10”, “Jack”, “Queen”, “King”}

Suits = {“Hearts”, “Spades”, “Clubs”, “Diamonds”}

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

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

1. Yes, the index for ranks must match up with the index for suits

Activity 3

1. (in Shuffler calss)
2. (in class)
3. 0 1 1 0

Activity 5

1. Constructor does not initialize size variable to cards.size()
2. Deck constructor initializes size wrong or does not initialize size
3. testShuffle method does not call the shuffle method so both decks are the same
4. deal method could deal the wrong index value card or the deck constructor initializes the deck with null values

Activity 6

1. 5 and 6,
2. Yes, the only combo that includes 3 cards is J, Q, and K. if all other combos are found the only 3 card combo left must be J, Q and K.
3. No, this game does not require strategy. You can never predict what card you are going to get next so you cannot plan out future moves.

Activity 7

1. Desk, deck of cards.

Private instance var of ElevensBoard class: size of board, cards in board, deck, suits pointvalues and ranks of cards.

1. When card is selected:

if legal

remove cards and replace

if no cards left win game

if no moves available

lose game

1. Yes
2. a. In the constructor and the new game method

b. isLegal, anotherPlayPossible

c. returned list = {2, 4, 5, 8}

d. for(int I = 0; I < cIndexes.size(); i++)

{

System.out.print(cIndexes.get(i) + “, “);

}

e. isLegal, if an empty slot is selected it has to know

Activity 8

1. All games are played on boards and have the same win condition and deal method. They all have the method isLegal and anotherPlayPossible but they each have different code for each class.
2. The Elevens board inherits the board classes instance variables and can initialize them with super(vars). This is called
3. Yes, all differences between the classes are abstract methods and all common methods are implemented.

Activity 9

1. Size can be initialized in elevens or thirteens board in the constructor, size is not initialized in a method.
2. The conditions in which the cards can be removed is different for each game so it has to be implemented in the elevensboard or thirteensboard
3. An interface would allow it to call the methods polymorphically because each different board (elevens, thirteens) would have to implement different methods for islegal and anotherplaypossible. This is worse than using an abstract class because you would have to reimplement common methods and instance vars for each different board.

Activity 11

1.) wins ranged from 0 - 2 wins. They usually stay at 0 or 1 and sometimes goes to 2 wins.

2.) wins ranges from 10 - 16. They are more consistent.

3.) once I changes games played to 10,000 it almost always gave 10.x percent