Activity 2-

1. A deck contains multiple cards.
2. 3
3. String[] ranks = {“2”…"ace"};

String[] suits = {"clubs”,”spades”,”hearts”,”diamonds”};

int[] pointValues = {2…11};

1. For rank and pointvalue yes it does.

Activity 3-

1. In bluejay
2. In bluejay
3. 0,1,1,0

Activity 6-

1. The 6 and either 5
2. Yes, each card can only be matched with one other to equal 11 and there are an even number of cards to be matched therefore it is impossible to have 3 left that aren’t J,Q,K
3. The only strategic move to make is to make matches that get rid a cards you have two of, all the rest is up to chance

Activity 7-

1. Standard deck of playing cards, a human, and a table or desk

ElevensBoard – BOARD\_SIZE,RANKS,SUITS,POINT\_VALUES,cards,deck,I\_AM\_DEBUGGING

1. Shuffle deck

Deal 9 cards

While deck isn’t empty

Remove cards specified by player

Deal new cards

1. No, it doesn’t have any event handler if the player lost
2. newGame method
3. anotherPlayIsPossible method
4. 0,1,3,6,7,8
5. In bluejay
6. anotherPlayIsPossible because it is after the player makes a move removing cards

Activity 8-

1. similarities: they all use a standard deck of 52 cards 2-ace four different suits, they are essentially played the same way by removing cards that add up to some number, and the goal is to have no cards left on the board

differences: the sum of cards removed is different in all three, the number of cards on the board at one time is also different, and the face cards behave differently in all three

1. ElevensBoard is a subclass of Board so all it needs to do is call the super method to run the constructor of the parent class initializing all those instance variables
2. The abstract methods are not implemented in ElevensBoard

Activity 9-

1. The size is initialized in the constructor so you only need a constant in each subclass of Board
2. The array cards doesn’t represent the cards that are currently on the board
3. It would work and it wouldn’t be as effective because there are a lot of methods that you would need to define in every subclass