Name _____ Role (circle one) programmer/computer/project manager

Name _____ Role (circle one) programmer/computer/project manager

Name _____ Role (circle one) programmer/computer/project manager

Name _____ Role (circle one) quality control

Card Counter

Your Tasks (Mark these off as you go)

- ☐ Create the CardCounter and CardPanel classes
- ☐ Write the CardCounter constructor
- ☐ Write the CardPanel class
- ☐ Access the dealtCards array in the CardDealer class
- ☐ Have Ms. Pluska check off your CardCounter and CardPanel classes
- ☐ Complete challenges 1 thru 3
- ☐ Have Ms. Pluska check off your challenges 1 thru 3 before you continue
- ☐ Receive credit for the group portion of this lab
- ☐ Receive credit for the individual portion of this lab

☐ Create the CardCounter and CardPanel classes

Locate the two large sheets of paper at your assigned location

On the first sheet of paper,

- Write "CardCounter class" at the top of the page

The CardCounter class will use the methods we created in the CardDealer class previously. To access these methods we will use the keyword "extends" to indicate that the CardCounter is a child of the CardDealer parent.

- Declare the CardCounter class using the appropriate signature

On the second sheet of paper,

- Write "CardPanel class" at the top of the page
- Declare the "CardPanel" class using the appropriate signature.

Your papers should like the example below,

Sheet 1	Sheet 2

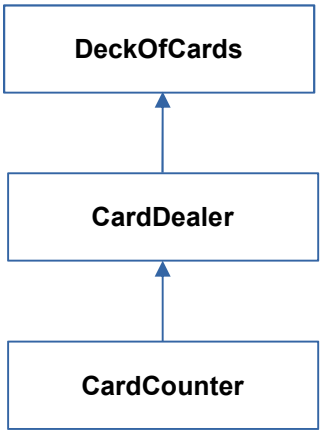
CardCounter class
`public class CardCounter extends CardDealer{`
 	
`//Leave lots of space here`
 	
`}`
	CardPanel class
	`public class CardPanel{`
	
	`//Leave lots of space here`
	
	`}`

☐ Write the CardCounter constructor

The purpose of the `CardCounter` class is to count cards in a given hand. For example, the number of cards in each suite, or the number of cards of a certain value.

Counting the cards will require methods from some other classes: Card, DeckOfCards, and CardDealer

As aforementioned, to access this information CardCounter will "extend" the CardDealer class. And likewise, the CardDealer class will "extend" the DeckOfCards class. The final hierarchy of our program will look as follows,



The signature for the CardDealer constructor is as follows,

```
public CardDealer(int d)
```

Because the CardDealer constructor is expecting a parameter, we will need to pass the information required of the CardDealer constructor in the CardCounter constructor. The keyword "super" will be used to create an instance of the parent, CardDealer.

CardCounter constructor

```
public CardCounter(int d){
        super(d);
}
```

□ Write the CardPanel class

In the CardPanel class you will create a new CardCounter object. Notice the CardCounter constructor is expecting a parameter of type int. This parameter will represent the size of the deal.

CardPanel class

```
public class CardPanel{

   public CardPanel(){

        dealSize = 10;
        CardCounter myDeal = new CardCounter(dealSize);
}
```

☐ Access the dealtCards array in the CardDealer class

Because CardCounter class extends the CardDealer class, the CardCounter can access all the public methods and variables. To get started counting the cards, the CardCounter needs to know which cards have been dealt by the dealer,

We can access the dealt cards by calling the getDealtCards() method,

```
myDeal.getDealtCards()
```

This method returns the array of dealt cards we need. To manipulate this array further we will assign it to a Card array which we will declare in the CardCounter class,

```
Card cardArray[];
```

Your final CardPanel class should look as follows,

Completed CardPanel class

```
public class CardPanel{

   private Card cardArray[];
   public CardPanel(){

        dealSize = 10;
        CardCounter myDeal = new CardCounter(dealSize);
        cardArray = myDeal.getDealtCards();
}
```

☐ Have Ms. Pluska check off your CardCounter and CardPanel classes before you continue

Before you continue have Ms. Pluska check off your CardCounter and CardPanel classes

Do not continue until you have Ms. Pluska's (or her designated TA's) signature _____

☐ Complete Challenges 1 thru 3

Challenge 1

In the CardCounter class, write a method sumDeal. This method should return an int which represents the sum of all the dealt cards. Because the CardCounter extends the CardDealer, you do not need a parameter to access the dealtCards array. Simply call getDealtCards() to access this array.

Call this method in the CardPanel class and print the result

Challenge 2

In the CardCounter class, write a method called countValues that finds the number of aces, twos, threes, etc. This method should return an array of the stored.

Call this method in the CardPanel class and print the result

Challenge 3

In the CardCounter class, write a method called countSuites that finds the number of cards in each suite.

Call this method in the CardPanel class and print the result

Have Ms. Pluska check off challenges 1 thru 3

Before you continue have Ms. Pluska check off challenges 1 thru 3.

Do not continue until you have Ms. Pluska's (or her designated TA's) signature _____

☐ Receive Credit for the group portion of this lab

Make sure to indicate the names of all group members, then submit this lab to the needs to be graded folder to receive credit for the group portion of this lab.

☐ Receive Credit for the individual portion of this lab

Implement challenges 1 thru 3 on your computer. Show Ms. Pluska the completed challenges to receive credit for the individual portion of this lab.