```
In [ ]: ### Template
        import random

        class Card:
            def __init__(self, value, suite):
                self.value = value
                self.suite = suite

            def __str__(self):
                return f"{self.value} of {self.suite}"

            def __eq__(self, other):
                """Check if two cards are the same"""
                # -- YOUR CODE HERE --
                #compare value then compare suite
                #if(self.value==other.value and self.suite==other.suite):
                    #then they are equal
                return(self.value==other.value and self.suite==other.suite)


        class CardSet:
            def __init__(self):
                self.cards = []

            def view(self):
                for card in self.cards:
                    print(card)

            def add_cards(self, cards):
                """Add cards to your set"""
                # -- YOUR CODE HERE --
                #use append
                self.cards.append(cards)

        class Deck(CardSet):
            def __init__(self):
                """Initialize the 52-card set. Start from 1-10, then Jack, Queen, King, then by suite: clubs, spades, hearts, diamonds"""
                cards = []
                # -- YOUR CODE HERE --
                #declare list of values
                value=['1', '2', '3', '4', '5', '6', '7', '8', '9','10', 'Jack', 'Queen', 'King']
                suite=["clubs","spades","hearts","diamonds"]

                #generate list of 52 cards
                #declare empty list
                cards=[]
                card_count=1
                for j in suite:
                    for i in value:

                        #declare the name variable
                        name="Card"+str(card_count)

                        #iterate the card counter
                        card_count+=1

                        #use locals() function to make dynamic variable names instead of manually creating 52 card objects
                        locals()[name]=Card(i,j)

                        #add to the list
                        cards.append(locals()[name])

                        #
                        #remarks:
                        #I realize now that I did not need to use variable names to add the Card Object to the list
                        #For now I will keep it since it does not break any of the functions
                        #


                self.cards = cards

            def count_cards(self):
                print("Cards Left: "+ str(len(self.cards)))

            def shuffle(self, seed=None):
                random.seed(seed)
                random.shuffle(self.cards)

            def peek(self, number=5):
                """Show the top n cards of the stack. This is analogous to getting the last n cards then reversing it."""
                # -- YOUR CODE HERE --

                #Recall: "Top of the cards" refers to the end of the list
                #Therefore "Top card"=cards[-1]

                #add if-case for when cards are less than 5
                if(len(self.cards)<5):
                    number=len(self.cards)


                #initialize counter and list
                p_count=0
                peak=[]

                #while loop keeps count of how many cards is to be peaked
                while p_count<number:
                    p_count+=1

                    #the end of the list will be appended to peak[]
                    #iterates backwards up to n times
                    peak.append(self.cards[p_count*-1])

                #print the top n cards
                for i in peak: print(i)

                #return peak list
                return(peak)



            def draw(self, cardset, number=5):
                """Transfer the top n cards of the stack to your cardset."""
                # -- YOUR CODE HERE --
                #Solution: use peek solution, but append() and remove() from the list as necessary

                #add if-case for when cards are less than 5
                if(len(self.cards)<5):
                    number=len(self.cards)

                #initialize counter and list
                p_count=0

                #while loop keeps count of how many cards is to be peaked
                while p_count<number:
                    p_count+=1

                    #the end of the list will be added using object function
                    #because of removal, index [-1] can simply be repeated every loop
                    cardset.add_cards(self.cards.pop(-1))



            def add_cards(self):
                pass

        if __name__ == "__main__":
            seed, hand, peek = input().split(",")

            myDeck = Deck()
            handA = CardSet()
            handB = CardSet()

            myDeck.shuffle(int(seed))

            for x in range(1,3):
                print(f"\nRound {x}:")

                myDeck.draw(handA, int(hand))
                myDeck.draw(handB, int(hand))

                print("Hand A: ")
                handA.view()
                print("Hand B: ")
                handB.view()

                myDeck.count_cards()
                if(x == 1):
                    print(f"\n{peek} Cards at the top: ")
                    myDeck.peek(int(peek))
```