In [5]:
```
import random



suits={'Hearts','Diamonds','Spades','Clubs'}

ranks={'Two','Three','Four','Five','Six','Seven','Eight','Nine','Ten','Jack','Quee

values={'Two':2,'Three':3,'Four':4,'Five':5,'Six':6,'Seven':7,'Eight':8,'Nine':9,'
        'Jack':10,'Queen':10,'King':10,'Ace':11}
```

In [6]:
```
class card():

    def __init__(self,suit,rank):

        self.suit=suit
        self.rank=rank
        self.values=values[rank]

    def __str__(self):

        return f"{self.rank} of {self.suit}"
```

In [7]:
```
new_card=card('Hearts','Three')
```

In [8]:
```
new_card.rank
```

Out[8]:
```
'Three'
```

In [9]:
```
new_card.suit
```

Out[9]:
```
'Hearts'
```

In [10]:
```
new_card.values
```

Out[10]:
```
3
```

In [11]:
```
print(new_card)
```

```
Three of Hearts
```

In [12]:
```
class Deck():

    def __init__(self):

        self.all_cards=[]

        for suit in suits:

            for rank in ranks:

                created_card=card(suit,rank)
                self.all_cards.append(created_card)

    def __str__(self):

        print(self.all_cards)

    def shuffle(self):
        random.shuffle(self.all_cards)

    def deal(self):

        return self.all_cards.pop(0)
```

In [13]:
```
mycard=Deck()
```

In [14]:
```
class Player_Hand():

    def __init__(self):

        self.cards=[]
        self.value=0
        self.aces=0

    def add_card(self,card):


        self.cards.append(card)

        self.value += values[card.rank]

    def adjust_for_ace(self):

            if self.value>17:

                self.cards.append(Chosen_card)

                self.value += 1

            if self.value<17:

                self.cards.append(Chosen_card)

                self.value += 11
```

In [15]:
```
player_hand=Player_Hand()
```

In [16]:
```
class Dealer_Hand():

    def __init__(self):

        self.cards=[]
        self.value=0
        self.aces=0

    def add_card(self,card):


        self.cards.append(card)

        self.value += values[card.rank]

    def adjust_for_ace(self):

            if self.value>17:
                self.cards.append(Chosen_card)

                self.value += 1

            if self.value<17:

                self.cards.append(Chosen_card)

                self.value += 11
```

In [17]:
```
dealer_hand=Dealer_Hand()
```

In [18]:
```
class Player_chips():

    def __init__(self):

        self.total=100
        self.bet=0

    def win_bet(self):

        self.total += self.bet

    def lose_bet(self):

        self.total -= self.bet
```

In [19]:
```
player_chips=Player_chips()
```

In [20]:
```
class Dealer_chips():

    def __init__(self):

        self.total=100
        self.bet=0

    def win_bet(self):

        self.total += self.bet

    def lose_bet(self):

        self.total -= self.bet
```

In [21]:
```
dealer_chips=Dealer_chips()
```

In [22]:
```
def take_bet():

    while True:

        try:

            player_chips.bet=int(input('Please enter a bet value: '))
            print(f"The bet amount is {player_chips.bet}")
            break

        except:
             print("That's an invalid input!Please enter a correct input")
```

In [23]:
```
take_bet()
```

```
Please enter a bet value: 6
The bet amount is 6
```

In [43]:
```
def hit(deck,hand):
    
    if player_hand.value<17:
        
        player_hand.add_card(mycard.deal())
```

In [44]:
```
def hit_or_stand(deck,hand):
    
    global playing
    
    while True:
        
        choice=input("Do you want to hit('x') or stand('s')?: ")
        
        if choice=='x':
            
            hit(deck,hand)
            
        if choice=='s':
            
            print("Player opts to stand. Dealer plays")
            playing=False
            
        break
```

In [45]:
```
def show_some():
    print("Dealer's cards:")
    print("<card hidden>")
    print(dealer_hand.cards[1])
    print("\n")
    print("Player's cards:")
    print(player_hand.cards[0])
    print(player_hand.cards[1])
    print("\n")

def show_all():
    print("Dealer's cards:")
    print(*dealer_hand.cards,sep='\n')
    
    print("\n")
    print("Player's cards:")
    print(*player_hand.cards,sep='\n')
    print("\n")
```

In [46]:
```
def player_busts():
    
    
        print("Player Busts")
        player_chips.lose_bet()
        
        
def player_wins():
        print("Player wins")
        player_chips.win_bet()

def dealer_busts():


        print("Dealer Busts")
        dealer_chips.lose_bet()


def dealer_wins():


        print("Dealer wins")
        dealer_chips.win_bet()

def push():
    print("Dealer and Player tie! It's a push.")
```

In [47]:
```
from IPython.display import clear_output
```

In [48]:
```
gameon=True

playing=True

while gameon==True:

    clear_output()

    print("Welcome to Blackjack!")

    mycard=Deck()
    mycard.shuffle()

    dealer_chips=Dealer_chips()
    player_chips=Player_chips()

    take_bet()

    player_hand=Player_Hand()
    player_hand.add_card(mycard.deal())
    player_hand.add_card(mycard.deal())

    dealer_hand=Dealer_Hand()
    dealer_hand.add_card(mycard.deal())
    dealer_hand.add_card(mycard.deal())

    show_some()

    show_all()

    while playing:

        hit_or_stand(mycard,player_hand)

        show_all()

        if player_hand.value>=21:

            player_busts()

            break
        if player_hand.value>dealer_hand.value and player_hand.value!=21:

            player_wins()

            break

        if dealer_hand.value>=21:

            dealer_busts()

            break

        if player_hand.value<dealer_hand.value and dealer_hand.value!=21:

            dealer_wins()

            break

    print(f"Player's winnings stand at {player_chips.total}")

    result=input("Do you want to continue?(y or n): ")

    if result=="y":

        gameon =True

    if result=="n":

        gameon=False
```

```
Welcome to Blackjack!
Please enter a bet value: 7
The bet amount is 7
Dealer's cards:
<card hidden>
Seven of Spades


Player's cards:
Three of Diamonds
Five of Clubs


Dealer's cards:
Six of Diamonds
Seven of Spades


Player's cards:
Three of Diamonds
Five of Clubs


Do you want to hit('x') or stand('s')?: x
Dealer's cards:
Six of Diamonds
Seven of Spades


Player's cards:
Three of Diamonds
Five of Clubs
King of Spades


Player wins
Player's winnings stand at 107
Do you want to continue?(y or n): n
```