In [14]:
```
import random

suits = ('Hearts', 'Diamonds', 'Spades', 'Clubs')
ranks = ('Two', 'Three', 'Four', 'Five', 'Six', 'Seven', 'Eight', 'Nine'
, 'Ten', 'Jack', 'Queen', 'King', 'Ace')
values = {'Two':2, 'Three':3, 'Four':4, 'Five':5, 'Six':6, 'Seven':7, 'E
ight':8, 'Nine':9, 'Ten':10, 'Jack':10,
          'Queen':10, 'King':10, 'Ace':11}

playing = True
```

In [15]:
```
class Card:

    def __init__(self,suit,rank):
        self.suit=suit
        self.rank=rank

    def __str__(self):
        return "{} of {}".format(self.rank,self.suit)
```

In [16]:
```
class Deck:

    def __init__(self):
        self.deck = []  # start with an empty list
        for suit in suits:
            for rank in ranks:
                self.deck.append(Card(suit,rank))  # build Card objects
 and add them to the list

    def __str__(self):
        deck_comp = ''  # start with an empty string
        for card in self.deck:
            deck_comp += '\n '+card.__str__() # add each Card object's p
rint string
        return 'The deck has:' + deck_comp

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

    def deal(self):
        single_card = self.deck.pop()
        return single_card
```

In [17]:
```
test_deck = Deck()
print(test_deck)
```

```
The deck has:
 Two of Hearts
 Three of Hearts
 Four of Hearts
 Five of Hearts
 Six of Hearts
 Seven of Hearts
 Eight of Hearts
 Nine of Hearts
 Ten of Hearts
 Jack of Hearts
 Queen of Hearts
 King of Hearts
 Ace of Hearts
 Two of Diamonds
 Three of Diamonds
 Four of Diamonds
 Five of Diamonds
 Six of Diamonds
 Seven of Diamonds
 Eight of Diamonds
 Nine of Diamonds
 Ten of Diamonds
 Jack of Diamonds
 Queen of Diamonds
 King of Diamonds
 Ace of Diamonds
 Two of Spades
 Three of Spades
 Four of Spades
 Five of Spades
 Six of Spades
 Seven of Spades
 Eight of Spades
 Nine of Spades
 Ten of Spades
 Jack of Spades
 Queen of Spades
 King of Spades
 Ace of Spades
 Two of Clubs
 Three of Clubs
 Four of Clubs
 Five of Clubs
 Six of Clubs
 Seven of Clubs
 Eight of Clubs
 Nine of Clubs
 Ten of Clubs
 Jack of Clubs
 Queen of Clubs
 King of Clubs
 Ace of Clubs
```

In [18]:
```
class Hand:
    def __init__(self):
        self.cards = []  # start with an empty list as we did in the Dec
k class
        self.value = 0   # start with zero value
        self.aces = 0    # add an attribute to keep track of aces

    def add_card(self,card):
        self.cards.append(card)
        self.value += values[card.rank]
        #if we have an ace
        if card.rank=="Ace":
            self.aces+=1

    def adjust_for_ace(self):
        #if we have an ace,by default it will be added as 21
        #if our total val is > 21 then instead of 11 use 1 i.e the total
value-10
        #and remove an ace  -1
        while self.value>21 and self.aces:
            self.value-=10
            self.aces-=1
```

In [19]:
```
test_deck=Deck()
test_deck.shuffle()
test_player=Hand()
pulled_card=test_deck.deal()
print(pulled_card)
test_player.add_card(pulled_card)
print(test_player.value)
```

```
Seven of Spades
7
```

In [20]:
```
class Chips:

    def __init__(self):
        self.total = 100  # This can be set to a default value or suppli
ed by a user input
        self.bet = 0

    def win_bet(self):
        self.total+=self.bet

    def lose_bet(self):
        self.total-=self.bet
```

In [21]:
```
def take_bet(chips):
    while True:
        try:
            chips.bet=int(input("Decide your bet chips: "))
        except:
            print("Sorry,give a proper value!")
            continue
        else:
            if chips.bet>chips.total:
                print("sorry,you dont have enough chips.You have {}".for
mat(chips.total))
            else:
                break
```

In [22]:
```
def hit(deck,hand):
    hand.add_card(deck.deal())
    hand.adjust_for_ace()
```

In [23]:
```
def hit_or_stand(deck,hand):
    global playing  # to control an upcoming while loop
    while True:
        x=input("Enter if you want to hit or stand?(if hit enter h and i
f stand enter s):")
        if x[0].lower()=="h":
            hit(deck,hand)
        elif x[0].lower()=="s":
            print("Player stands and now it is dealers turn")
            playing=False
        else:
            print("Enter correctly")
            continue
        break
```

In [24]:
```
def show_some(player,dealer):
    print("DEALERS HAND:")
    print("one card hidden!")
    print(dealer.cards[1])
    print('\n')
    print("PLAYERS HAND")
    for card in player.cards:
        print(card)
def show_all(player,dealer):
    print("DEALERS HAND")
    for card in dealer.cards:
        print(card)
    print("\n")
    print("PLAYERS HAND")
    for card in player.cards:
        print(card)
```

In [25]:
```
def player_busts(player,dealer,chips):
    print("BUST PLAYER")
    chips.lose_bet()

def player_wins(player,dealer,chips):
    print("PLAYER WON")
    chips.win_bet()

def dealer_busts(player,dealer,chips):
    print("BUST DEALER")
    chips.lose_bet()

def dealer_wins(player,dealer,chips):
    print("DEALER WON")
    chips.win_bet()

def push(player,dealer):
    print("IT'S A TIE")
```

In [ ]:
```
while True:
    # Print an opening statement
    print("WELCOME TO BLACKJACK")
    # Create & shuffle the deck, deal two cards to each player
    deck=Deck()
    deck.shuffle()
    player_hand=Hand()
    player_hand.add_card(deck.deal())
    player_hand.add_card(deck.deal())

    dealer_hand=Hand()
    dealer_hand.add_card(deck.deal())
    dealer_hand.add_card(deck.deal())
    # Set up the Player's chips
    player_chips=Chips()
    # Prompt the Player for their bet
    take_bet(player_chips)
    # Show cards (but keep one dealer card hidden)
    show_some(player_hand,dealer_hand)
    while playing:  # recall this variable from our hit_or_stand functio
n
        # Prompt for Player to Hit or Stand
        hit_or_stand(deck,player_hand)
        # Show cards (but keep one dealer card hidden)
        show_some(player_hand,dealer_hand)
        # If player's hand exceeds 21, run player_busts() and break out
of loop
        if player_hand.value>21:
            player_busts(player_hand,dealer_hand,player_chips)
            break
    # If Player hasn't busted, play Dealer's hand until Dealer reaches 1
7
    if player_hand.value <= 21:
        while dealer_hand.value<player_hand.value:
            hit(deck,dealer_hand)
            # Show all cards
        show_all(player_hand,dealer_hand)
        # Run different winning scenarios
        if dealer_hand.value>21:
            dealer_busts(player_hand,dealer_hand,player_chips)
        elif dealer_hand.value>player_hand.value:
            dealer_wins(player_hand,dealer_hand,player_chips)
        elif dealer_hand.value<player_hand.value:
            payer_wins(player_hand,dealer_hand,player_chips)
        else:
            push(player_hand,dealer_hand)
    # Inform Player of their chips total
    print(f"n The total chips of player are {}".format(player_chips.tota
l))
    # Ask to play again
    again=input("DO YOU WANT TO PLAY AGAIN?y/n:")
    if again[0].lower()=="y":
        playing=True
        continue
    else:
        print("THANK YOU FOR PLAYING")
        break
```

```
WELCOME TO BLACKJACK
Decide your bet chips: 100
DEALERS HAND:
one card hidden!
Jack of Clubs


PLAYERS HAND
Two of Clubs
Ace of Spades
Enter if you want to hit or stand?(if hit enter h and if stand enter s):
h
DEALERS HAND:
one card hidden!
Jack of Clubs


PLAYERS HAND
Two of Clubs
Ace of Spades
Eight of Diamonds
Enter if you want to hit or stand?(if hit enter h and if stand enter s):
h
DEALERS HAND:
one card hidden!
Jack of Clubs


PLAYERS HAND
Two of Clubs
Ace of Spades
Eight of Diamonds
Seven of Hearts
Enter if you want to hit or stand?(if hit enter h and if stand enter s):

Player stands and now it is dealers turn
DEALERS HAND:
one card hidden!
Jack of Clubs


PLAYERS HAND
Two of Clubs
Ace of Spades
Eight of Diamonds
Seven of Hearts
DEALERS HAND:
Ace of Hearts
Jack of Clubs


PLAYERS HAND
Two of Clubs
Ace of Spades
Eight of Diamonds
Seven of Hearts
DEALER WON
/n The total chips of player are 200
DO YOU WANT TO PLAY AGAIN?y/n:y
WELCOME TO BLACKJACK
Decide your bet chips: 100
DEALERS HAND:
one card hidden!
Five of Diamonds


PLAYERS HAND
Ten of Clubs
Five of Spades
Enter if you want to hit or stand?(if hit enter h and if stand enter s):
h
Player stands and now it is dealers turn
DEALERS HAND:
one card hidden!
Five of Diamonds


PLAYERS HAND
Ten of Clubs
Five of Spades
DEALERS HAND
Five of Clubs
Five of Diamonds
Ace of Diamonds


PLAYERS HAND
Ten of Clubs
Five of Spades
DEALER WON
/n The total chips of player are 200
DO YOU WANT TO PLAY AGAIN?y/n:y
WELCOME TO BLACKJACK
Decide your bet chips: 12
DEALERS HAND:
one card hidden!
Eight of Diamonds


PLAYERS HAND
Four of Hearts
Seven of Hearts
Enter if you want to hit or stand?(if hit enter h and if stand enter s):
h
DEALERS HAND:
one card hidden!
Eight of Diamonds


PLAYERS HAND
Four of Hearts
Seven of Hearts
Nine of Clubs
Enter if you want to hit or stand?(if hit enter h and if stand enter s):

Player stands and now it is dealers turn
DEALERS HAND:
one card hidden!
Eight of Diamonds


PLAYERS HAND
Four of Hearts
Seven of Hearts
Nine of Clubs
DEALERS HAND
Eight of Diamonds
Ace of Clubs


PLAYERS HAND
Four of Hearts
Seven of Hearts
Nine of Clubs
DEALER WON
/n The total chips of player are 212
DO YOU WANT TO PLAY AGAIN?y/n:n
THANK YOU FOR PLAYING
```