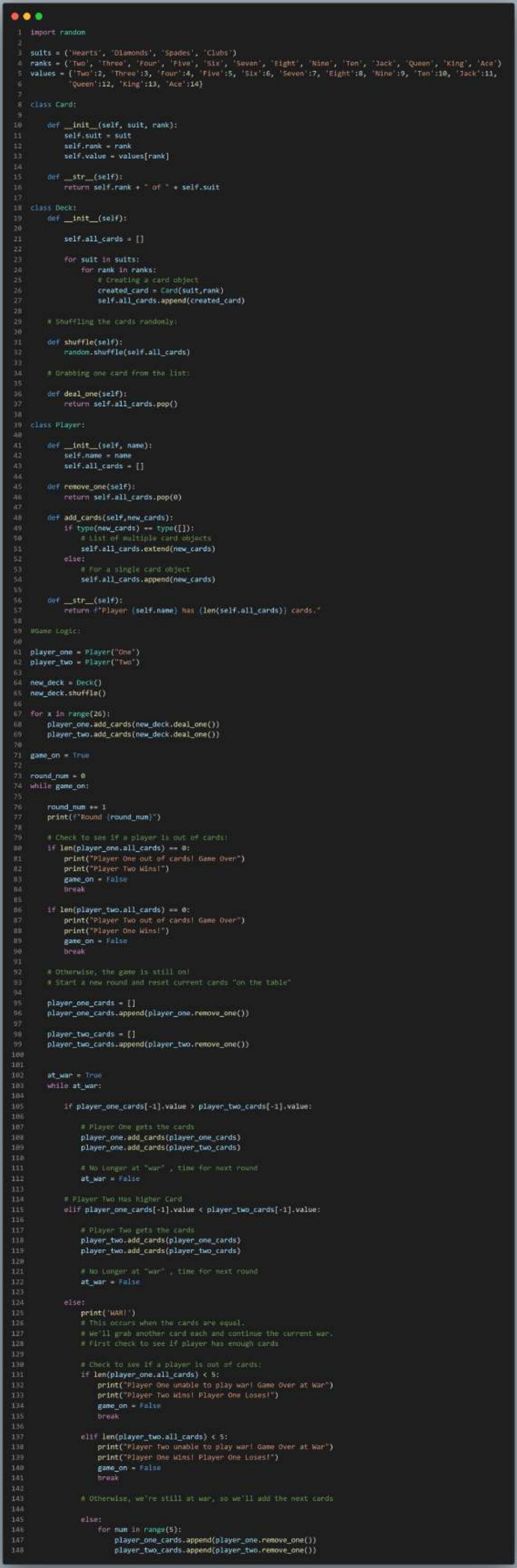

```python
import random

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, 'Eight':8, 'Nine':9, 'Ten':10, 'Jack':11,
         'Queen':12, 'King':13, 'Ace':14}

class Card:

    def __init__(self, suit, rank):
        self.suit = suit
        self.rank = rank
        self.value = values[rank]

    def __str__(self):
        return self.rank + " of " + self.suit

class Deck:
    def __init__(self):

        self.all_cards = []

        for suit in suits:
            for rank in ranks:
                # Creating a card object
                created_card = Card(suit,rank)
                self.all_cards.append(created_card)

    # Shuffling the cards randomly:

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

    # Grabbing one card from the list:

    def deal_one(self):
        return self.all_cards.pop()

class Player:

    def __init__(self, name):
        self.name = name
        self.all_cards = []

    def remove_one(self):
        return self.all_cards.pop(0)

    def add_cards(self,new_cards):
        if type(new_cards) == type([]):
            # List of multiple card objects
            self.all_cards.extend(new_cards)
        else:
            # For a single card object
            self.all_cards.append(new_cards)

    def __str__(self):
        return f"Player {self.name} has {len(self.all_cards)} cards."

#Game Logic:

player_one = Player("One")
player_two = Player("Two")

new_deck = Deck()
new_deck.shuffle()

for x in range(26):
    player_one.add_cards(new_deck.deal_one())
    player_two.add_cards(new_deck.deal_one())

game_on = True

round_num = 0
while game_on:

    round_num += 1
    print(f"Round {round_num}")

    # Check to see if a player is out of cards!
    if len(player_one.all_cards) == 0:
        print("Player One out of cards! Game Over")
        print("Player Two Wins!")
        game_on = False
        break

    if len(player_two.all_cards) == 0:
        print("Player Two out of cards! Game Over")
        print("Player One Wins!")
        game_on = False
        break

    # Otherwise, the game is still on!
    # Start a new round and reset current cards "on the table"

    player_one_cards = []
    player_one_cards.append(player_one.remove_one())

    player_two_cards = []
    player_two_cards.append(player_two.remove_one())


    at_war = True
    while at_war:

        if player_one_cards[-1].value > player_two_cards[-1].value:

            # Player One gets the cards
            player_one.add_cards(player_one_cards)
            player_one.add_cards(player_two_cards)

            # No Longer at "war" , time for next round
            at_war = False

        # Player Two Has higher Card
        elif player_one_cards[-1].value < player_two_cards[-1].value:

            # Player Two gets the cards
            player_two.add_cards(player_one_cards)
            player_two.add_cards(player_two_cards)

            # No Longer at "war" , time for next round
            at_war = False

        else:
            print('WAR!')
            # This occurs when the cards are equal.
            # We'll grab another card each and continue the current war.
            # First check to see if player has enough cards

            # Check to see if a player is out of cards:
            if len(player_one.all_cards) < 5:
                print("Player One unable to play war! Game Over at War")
                print("Player Two Wins! Player One Loses!")
                game_on = False
                break

            elif len(player_two.all_cards) < 5:
                print("Player Two unable to play war! Game Over at War")
                print("Player One Wins! Player One Loses!")
                game_on = False
                break

            # Otherwise, we're still at war, so we'll add the next cards

            else:
                for num in range(5):
                    player_one_cards.append(player_one.remove_one())
                    player_two_cards.append(player_two.remove_one())
```