```
# import necessary functions
from random import randint
from IPython.display import clear_output

# create the blackjack class, which will hold all game methods and attributes
class Blackjack():
    def __init__(self):
        self.deck = []    # set to an empty list
        self.suits = ("Spades", "Hearts", "Diamonds", "Clubs")
        self.values = (2, 3, 4, 5, 6, 7, 8, 9, 10, 'J', 'Q', 'K', 'A')

    # create a method that creates a deck of 52 cards, each card should be a tuple with a
    value and suit
    def makeDeck(self):
        for suit in self.suits:
            for value in self.values:
                self.deck.append((value, suit))    # ex: (7, "Hearts")

    # method to pop a card from deck using a random index value
    def pullCard(self):
        return self.deck.pop(randint(0, len(self.deck) - 1))

# create a class for the dealer and player objects
class Player():
    def __init__(self, name):
        self.name = name
        self.hand = []
        self.currency = 500

    def getCurrency(self):
        return self.currency

    def setCurrency(self, amount, won):
        '''
            Take in amount to be added or subtracted, the won parameter will
            handle whether or not the player won and should be added or subtracted.
        '''
        if won:
            self.currency += amount
        elif not won:
            self.currency -= amount

    # take in a tuple and append it to the hand
    def addCard(self, card):
        self.hand.append(card)

    # if not dealer's turn, then only show one of his cards, otherwise show all cards
    def showHand(self, dealer_start=True):
        print("\n{}".format(self.name))
        print("===========")

        for i in range(len(self.hand)):
            if self.name == 'Dealer' and i == 0 and dealer_start:
                print("- of -")  # hide first card
            else:
                card = self.hand[i]
                print("{} of {}".format(card[0], card[1]))
        print("Total = {}".format(self.calcHand(dealer_start)))

    # if not dealer's turn then only give back total of second card
    def calcHand(self, dealer_start=True):
        total = 0
        aces = 0   # calculate aces afterwards
        card_values = {1:1, 2:2, 3:3, 4:4, 5:5, 6:6, 7:7, 8:8, 9:9, 10:10, 'J':10, 'Q':10,
        'K':10, 'A':11}

        if self.name == 'Dealer' and dealer_start:
            card = self.hand[1]
            return card_values[card[0]]
```

```
        for card in self.hand:
            if card[0] == 'A':
                aces += 1
            else:
                total += card_values[card[0]]

        for i in range(aces):
            if total + 11 > 21:
                total += 1
            else:
                total += 11

        return total

game = Blackjack()
game.makeDeck()

name = input("What is your name?")
player = Player(name)
dealer = Player("Dealer")

# ask player how much they want to wager
wager = 0
print('You have ${}.'.format(player.getCurrency()))

while wager == 0 or player.getCurrency() - wager < 0:
    # handle getting and converting wager
    try:
        wager = int(input("How much would you like to wager? "))

        # if wager is too much print message
        if player.getCurrency() - wager < 0:
            print("Sorry you don't have that much money. Try again!")
    except:
        print('Something went wrong, please try again!')

# add two cards to the dealer and player hand
for i in range(2):
    player.addCard(game.pullCard())
    dealer.addCard(game.pullCard())

# show both hands using method
player.showHand()
dealer.showHand()

player_bust = False

while input('Would you like to stay or hit?').lower() != 'stay':
    clear_output()

    # pull card and put into player's hand
    player.addCard(game.pullCard())

    # show both hands using method
    player.showHand()
    dealer.showHand()

    # check if over 21
    if player.calcHand() > 21:
        player_bust = True
        break

# handling the dealer's turn, only run if player didn't bust
dealer_bust = False

if not player_bust:
    while dealer.calcHand(False) < 17:
        # pull card and put into player's hand
```

```
        dealer.addCard(game.pullCard())

        # check if over 21
        if dealer.calcHand(False) > 21:
            dealer_bust = True
            break

clear_output()

# show both hands using method
player.showHand()
dealer.showHand(False)

# calculate a winner
if player_bust:
    print('You busted, better luck next time!')
    player.setCurrency(wager, False)
elif dealer_bust:
    print('The dealer busted, you win!')
    player.setCurrency(wager, True)
elif dealer.calcHand(False) > player.calcHand():
    print('Dealer has higher cards, you lose!')
    player.setCurrency(wager, False)
elif dealer.calcHand(False) < player.calcHand():
    print('You beat the dealer! Congrats!')
    player.setCurrency(wager, True)
else:
    print('You pushed, no one wins!')

print('You ended with ${}.'.format(player.getCurrency()))
```