In [ ]:
```
#Millie White
#Blackjack Simulation
#3/12/2021
```

In [ ]:
```
import random
```

In [ ]:
```
#Create a deck of Cards
#Since only the number of the card matters in blackjack,
#The deck created will not differentiate different suits
#This code creates 1 deck of cards(#52 cards)
#Since blackjack is about suming up the cards to make 21,the cards with no number
#Ace is 1 ,jack is 11,queen is 12 and king is 13
#This is a global variable that can be accessed by all the functions
#n is the number of decks

n = 100 #number of decks
deck_of_cards = [1,2,3,4,5,6,7,8,9,10,11,12,13] * 4 * n # Four suits in a deck
```

In [ ]:
```
#Initialize variables to keep track of games won,lost and tied with dealer
number_of_games_won = 0
number_of_games_lost = 0
number_of_games_tied = 0
```

In [ ]:
```
#The deal funtion takes the shuffled deck and deals 2 cards each to the dealer ar

def deal(deck_of_cards):  #The function deal which takes a deck as an argument
    while len(deck_of_cards) >=1:
        hand=[] #initialize each hand as a list where cards are added in
        for i in range(2): #getting 2 cards for the blackjack game
            if len(deck_of_cards) >=1:
                random.shuffle(deck_of_cards)  #shuffle the deck
                card = deck_of_cards.pop() #remove last card from shuffled deck c

                if card == 1:
                    card = 'A'
                elif card == 11:
                    card = 'J'
                elif card == 12:
                    card = 'Q'
                elif card == 13:
                    card = 'K'

                hand.append(card)  #each hand now has 2 cards
        return hand
```

In [ ]:
```
#The sum_of_hand function add up the totals of the dealer and player hand to dete

def sum_of_hand(hand):
    total = 0
    for card in hand:
        if card == 'J' or card == 'Q' or card == 'K':
            total += 10
        elif card == 'A':
            if total >= 11:
                total += 1 #so that player doesn't go over 21
            else:
                total += 11
        else:
            total += card
    return total
```

In [ ]:
```
#The determine_winner function compares the totals of dealer and player hands and

def determine_winner(dealer,player):

    #Initialize variables to keep track of games won,lost and tied with dealer
    number_of_games_won = 0
    number_of_games_lost = 0
    number_of_games_tied = 0

    if (sum_of_hand(player) == 21 and sum_of_hand(dealer) < 21) or (sum_of_hand(d

        print("You win!")
        number_of_games_won += 1
        print("won",number_of_games_won)


    elif (sum_of_hand(dealer) == 21 and sum_of_hand(player) < 21) or (sum_of_hand

        print("Dealer wins!")
        number_of_games_lost += 1
        print("lost",number_of_games_lost)


    elif sum_of_hand(dealer) == 21 and sum_of_hand(player) == 21:

        print("Tie!")
        number_of_games_tied += 1
        print("ties",number_of_games_tied)


    return [number_of_games_tied, number_of_games_lost, number_of_games_won]
```

In [ ]:
```
#The hit function adds 1 more card to the dealer or player's hand

def hit(hand):

    if len(deck_of_cards) >=1:
        card = deck_of_cards.pop()
        if card == 1:
            card = 'A'
        if card == 11:
            card = 'J'
        if card == 12:
            card = 'Q'
        if card == 13:
            card = 'K'
        hand.append(card)
    return hand
```

In [ ]:
```
def main():

    #Initialize variables to keep track of games won,lost and tied with dealer
    number_of_games_won = 0
    number_of_games_lost = 0
    number_of_games_tied = 0

    #Introduce the player to the game
    print("This is a simulation of a black jack game between 1 dealer and 1 playe
    print('')



    #Initialize player decision to play as zero and then ask if they would like t
    interest_in_playing  = 'Y'
    #interest_in_playing = input("Would you like to play? 'Y' to continue and 'N
    if interest_in_playing == 'N':
        print('Bye!')



    #If the player chooses to play...
    while (interest_in_playing  != 'N'and len(deck_of_cards) >=1):  #As long as p



        #Deal 2 cards each to player and dealer
        #Initialize the variables player and dealer to keep track of the hands as
        dealer = deal(deck_of_cards)  #deal 2 cards to the dealer,one of the card
        player = deal(deck_of_cards)  #deal 2 cards to the player which are face

        #Check to see if there is a winner
        #printed_score(dealer,player)

        #If no winner,player will stand
        if sum_of_hand(dealer) == sum_of_hand(player):
            next_move = 'stand'

            if next_move == "stand" and len(deck_of_cards) >= 1:
                while sum_of_hand(dealer) < 21:  #if dealer has not reached 21
                    hit(dealer) #dealer gets another card
                    determine_winner(dealer,player)



        #Get a printed score of dealer and player hands and totals
        ANS = determine_winner(dealer,player)  #Determine if player wins,loses or
        #print(ANS)


        if ANS[0] != 0:
            number_of_games_tied += 1
        else:
            number_of_games_tied += 0
        print(number_of_games_tied)
```

```
        if ANS[1] != 0:
            number_of_games_lost += 1
        else:
            number_of_games_lost += 0
        print(number_of_games_lost)


        if ANS[2] != 0:
            number_of_games_won += 1
        else:
            number_of_games_won += 0
        print(number_of_games_won)


        print('tied',number_of_games_tied,'lost',number_of_games_lost,'won',numbe



        #Calculations
        total_number_of_games = number_of_games_tied + number_of_games_lost + num
        percent_won = (number_of_games_won / total_number_of_games ) * 100
        percent_lost = (number_of_games_lost / total_number_of_games ) * 100
        percent_tied = (number_of_games_tied / total_number_of_games ) * 100

        print('percentage of games won',percent_won)
        print('percentage of games lost',percent_lost)
        print('percentage of games tied',percent_tied)
```

In [ ]:
```
main()
```