In [1]:

```
import random

suits=('Hearts','Diamond','Spade','cluds')
ranks=('One','Two','Three','Four','Five','Six','Seven','Eight','Nine','Ten','Jack','Ki
values={'One':1,'Two':2,'Three':3,'Four':4,'Five':5,'Six':6,'Seven':7,'Eight':8,'Nine'
```

In [2]:

```
class Card:

    def __init__(self,suit,rank):
        self.suit=suit
        self.rank=rank
        self.value=values[rank]
    def __str__(self):
        return f'{self.rank} of {self.suit}'
```

In [3]:

```
mycard=Card('Hearts','One')
```

In [4]:

```
print(mycard)
```

```
One of Hearts
```

In [5]:

```
class Deck:

    def __init__(self):
        self.all_cards=[]
        for suit in suits:
            for rank in ranks:
                creat_card=Card(suit,rank)
                self.all_cards.append(creat_card)
    def shuffle(self):
        random.shuffle(self.all_cards)
    def deal_with_one(self):
        return self.all_cards.pop()
```

In [6]:

```
deck_one=Deck()
```

In [7]:

```
mycard=deck_one.deal_with_one()
mycard.value
```

Out[7]:

```
14
```

In [8]:

```
class Player:

    def __init__(self):
        self.all_cards=[]
    def add_on(self,card):
        if type(card)==type([]):
            self.all_cards.extend(card)
        else:

            self.all_cards.append(card)
    def __str__(self):
        return f'{self.name} has {len(self.all_cards)} cards'
```

In [9]:

```
myplayer=Player()
```

In [10]:

```
myplayer.add_on(mycard)
```

In [ ]:

In [13]:

```
#Game
print('Welcome to the game\n')
human_player=Player()
New_deck=Deck()
New_deck.shuffle()
computuer_delar=Player()
player1=[]
computer1=[]
new_bank=Bank()
sum_player=0
sum_com=0
while True:
    chips=int(input('How many chips do you want to bet'))
    if(chips<=new_bank.balance):
        break
    else:
        print("your are betting more than your balance" )

print('\nHuman player cards are:')
for mycard in range(2):

    mycard=New_deck.deal_with_one()
    print(mycard)
    player1.append(mycard)
    sum_palyer=sum_player+mycard.value
print('\nComputer palyer card is:')

com_card=New_deck.deal_with_one()
print(com_card)
computer1.append(com_card)
sum_com=sum_com+com_card.value
com_card=New_deck.deal_with_one()
print('<closed card>')
computer1.append(com_card)
sum_com=sum_com+com_card.value
game_on=True
print('\nHuman Player turn\n')

while True:
    print('Enter hit if you want to hit or Enter skip if you want to skip')
    result = input('Do you want to Hit')
    if result=='hit':
        mycard=New_deck.deal_with_one()
        player1.append(mycard)
        sum_player=sum_player+mycard.value
        if sum_player>21:
            print(f'Human Player sum is greater than 21 that is {sum_player}')
            print('Human Player lost the game')
            new_bank.win_game(chips)
            game_on=False
            break
    else:
        break
```

```
if game_on:
    print('computer turn\n')
    result2=random.randint(1,3)
    for i in range(result2):
        com_card=New_deck.deal_with_one()
        computer1.append(com_card)
        sum_com=sum_com+com_card.value
    if (sum_com>sum_player) and sum_com<21:
        print(f'Computer game is less than 21 {sum_com}')
        print('Computer won')
        new_bank.lost_game(chips)
    if sum_com>21:
        print(f'Computer player is greater than 21 that is {sum_com}')
        print('Human player won')
        new_bank.win_game(chips)

print('\ngame ends\nThanks for playing')
```

```
Welcome to the game

How many chips do you want to bet5

Human player cards are:
Ace of cluds
Five of Hearts

Computer palyer card is:
Four of Spade
<closed card>

Human Player turn

Enter hit if you want to hit or Enter skip if you want to skip
Do you want to Hithit
Enter hit if you want to hit or Enter skip if you want to skip
Do you want to Hitscape
computer turn

Computer player is greater than 21 that is 27
Human player won
player left with 105 chips

game ends
Thanks for playing
```

In [48]:

```
result=random.randint(1,3)
```

In [49]:

```
result
```

Out[49]:

```
2
```

In [12]:

```
class Bank:

    def __init__(self):
        self.balance=100
    def lost_game(self,chips):
        self.balance-=chips
        print(f'player left with {self.balance} chips' )
    def win_game(self,chips):
        self.balance+=chips
        print(f'player left with {self.balance} chips' )
```

In [ ]:

```
```

In [ ]:

```
```

In [ ]:

```
```