This will create a class of players that has the players name and the players BlackJack handValue which will be updated when they get a new card.

MAX = 21

Class Player:

#INPUT: self, name of player, hand value for that player

#PROCESS: bringing in name and hand value into the class using the conductor

#OUTPUT: none

def \_\_ init\_\_(self, n, hv):

name = input(“What is the first player’s name?”)

self.name = n

self.handValue = 0

#INPUT: self

#PROCESS: building a string of values to make it print prettier

#OUTPUT: none

def \_\_ str\_\_(self):

player1 = self.name1 + " your hand value is: " + str(self.handValue)

player2 = self.name2 + “ your hand value is: “ + str(self.handValue)

#INPUT: self

#PROCESS: updating the hand value for each player

#OUTPUT: the new hand value for each player

def update\_hand\_value(self):

if not card.startswith(‘Ace’):

self.handValue = self.handValue + value

elif handVal > 10:

self.handValue = self.handValue + value

else:

self.handValue = self.handValue + 11

return handValue

#INPUT: none

#PROCESS: calls functions

#OUTPUT: prints the player’s handvalues

def main():

instructions()

deck = createDeck()

while handValue1 <= MAX and handValue2 >= MAX:

card, value = deal(deck)

player1 = Player(name)

player2 = Player(name)

handValue1 = player1.update\_hand\_value()

handValue2 = player2.update\_hand\_value()

print(player)

#INPUT: None

#PROCESS: prints instructions for the game

#OUTPUT: the instructions

def instructions():

print(“This is how you play the game…”)

#INPUT: none

#PROCESS: This creates the dictionary with the deck of cards

#OUTPUT: none

def create\_Deck():

suits = [‘Spades’, ‘Hearts’, Clubs’, Diamonds’]

special\_values = {‘Ace’:1, ‘King’:10, ‘Queen’:10, ‘Jack’:10}

numbers = [‘Ace’, ‘King’, ‘Queen’, ‘Jack’]

for i in range(2,11):

numbers.append(str(i))

deck = {}

for suit in suits:

for num in numbers:

if num.isnumeric():

deck[num + ‘ of ‘ + suit] = int(num)

else:

deck[num + ‘ of ‘ + suit] = special\_values(num)

#INPUT: deck

#PROCESS: shuffles the keys randomly

#OUTPUT: The randomly selected key

def deal(deck):

keys = list(deck.keys())

random.shuffle(keys)

value = deck.pop(keys[0], value)

return(keys[0], value)