import random

class Card:

def \_\_init\_\_(self, suit, value):

self.suit = suit

self.value = value

def \_\_str\_\_(self):

return f"{self.value} of {self.suit}"

class Game:

suits = ['Spades', 'Hearts', 'Diamonds', 'Clubs']

values = [7, 8, 9, 10, 'Jack', 'Queen', 'King', 'Ace']

def \_\_init\_\_(self):

self.players = {i: [] for i in range(1, 5)}

self.deck = [Card(suit, value) for suit in Game.suits for value in Game.values]

def input\_cards(self):

for player in range(1, 5):

print(f"Player {player}, please enter your 13 cards.")

for i in range(13):

while True:

suit = input(f"Enter suit for card {i+1} (choose from {Game.suits}): ").capitalize()

if suit not in Game.suits:

print("Invalid suit. Try again.")

continue

value = input(f"Enter value for card {i+1} (choose from {Game.values}): ")

if value.isdigit():

value = int(value)

if value not in Game.values:

print("Invalid value. Try again.")

continue

elif value.capitalize() not in Game.values:

print("Invalid value. Try again.")

continue

else:

value = value.capitalize()

card = Card(suit, value)

self.players[player].append(card)

break

def shuffle\_and\_deal(self):

random.shuffle(self.deck)

for i, card in enumerate(self.deck):

self.players[(i % 4) + 1].append(card)

def show\_hands(self):

for player, hand in self.players.items():

print(f"Player {player}'s hand:")

for card in hand:

print(f" {card}")

print()

def play\_game(self):

choice = input("Do you want to enter cards manually? (yes/no): ").lower()

if choice == 'yes':

self.input\_cards()

else:

self.shuffle\_and\_deal()

self.show\_hands()

class AdvancedGame(Game):

def \_\_init\_\_(self):

super().\_\_init\_\_()

def calculate\_score(self):

scores = {player: sum([10 if isinstance(card.value, str) else card.value for card in hand]) for player, hand in self.players.items()}

return scores

def show\_scores(self):

scores = self.calculate\_score()

for player, score in scores.items():

print(f"Player {player} scored: {score}")

if \_\_name\_\_ == "\_\_main\_\_":

game = AdvancedGame()

game.play\_game()

game.show\_scores()