



Python OOP – Object-Oriented Programming for Beginners Mid-term Course Project Description



Goal:

- Create a text-based version of the Blackjack game (Twenty-One) using basic Object-Oriented Programming (OOP) principles.

Classes:

- **Card:**
 - Card instances are created by the Deck class.
 - **Instance Attributes:**
 - suit
 - Strings: either “Spades”, “Clubs”, “Diamonds”, “Hearts”.
 - value
 - Integers: from 1 to 11
 - **Methods:**
 - show():
 - Displays the suit and the value of a card instance using print().
- **Deck:**
 - **Class Attribute:**
 - suits
 - A list of the possible suits used to create card instances.
 - Strings: “Spades”, “Clubs”, “Diamonds”, “Hearts”.
 - **Instance Attributes:**
 - cards
 - A list of card instances.
 - Starts empty when the instance of Deck is created.
 - **Methods:**
 - build():
 - For every possible suit, create a card instance per value from 1 to 11.
 - show():
 - Calls the .show() method of all the cards in the list of cards of the deck.
 - shuffle():
 - Shuffle the deck using the Fisher-Yates Shuffle Algorithm.
 - draw():
 - Method to draw a card instance from the deck.
 - The method has to remove the card from the deck and return it.

- **Player:**
 - **Instance Attributes:**
 - name
 - hand
 - A list of card instances.
 - is_dealer
 - A boolean value. True if the player is the dealer. False otherwise.
 - By default, this is False.
 - **Methods:**
 - draw():
 - Parameter:
 - deck – The deck instance that the player will draw a card from.
 - This method appends the card that was drawn from the deck to the player's hand.
 - It returns self to allow method chaining.
 - show_hand():
 - Parameters:
 - reveal_card – False (by default) in case the card of the dealer shouldn't be revealed. True if the card of the dealer should be revealed. If the player that called the method is not the dealer, this value has no effect.
 - If the player is not the dealer, the .show() method of all the card instances in the hand should be called.
 - If the player is the dealer, the last card should be hidden.
 - discard():
 - This method removes the last card from the hand and returns it.
 - get_hand_value():
 - Adds the value of all the card instances of the hand and returns the sum.

- **CardGame:**

- The game starts automatically when an instance of CardGame is created.
- **Class Attributes:**
 - instructions – A detailed string that welcomes the player to the game and briefly presents the instructions.
- **Instance Attributes:**
 - deck – The deck instance associated with the instance of CardGame
 - Player – The player instance associated with the instance of CardGame.
 - Dealer – The player instance that is identified as the “dealer” associated with the instance of CardGame.
- **Methods:**
 - start_game():
 - The main logic of the game is contained in this method.
 - The number of turns should be displayed.
 - The deck is shuffled before the game starts.
 - The player and the dealer draw two cards when the game starts.
 - Then, as many times as needed:
 - The dealer’s hand is shown (hiding the last card).
 - The player’s hand is shown.
 - If total value of the hand is over 21, the player loses automatically and a descriptive message is printed.
 - A menu is displayed asking the user to choose if he/she would like to draw another card or stand.
 - Depending on the choice, the process stops or another round starts with the new hand.
 - If the player chooses to stand, the final values of the two hands are calculated (player and dealer).
 - The dealer’s hand is shown, revealing the value of the hidden card.
 - A descriptive message is printed telling the user if he/she won the game.
 - ask_choice():
 - Displays a menu with two options for the player:
 - 1 – Ask for another card
 - 2 – Stand
 - The user enters the number that corresponds to the option selected.
 - If an invalid value is entered, by default 2 is returned, so we assume that the player chose to stand.