This program has two enums for the suit and rank of a card. The Card class uses those enums to keep track of the identity of the card. It also has a value property, which represents the numeric value of the card. It is normally the same as the ordinal of the rank, with exceptions for king, queen, and jack.

The deck class is a wrapper around a LinkedList of cards. Its constructor initializes a deck of cards by adding a card to the LinkedLIst for every combination of rank and suite. It has a draw() method which returns a randomly chosen card from the LinkedList and removes it from the list, so that it will not be picked again.

The hand class represents a player’s cards. It also wraps around a LinkedList of cards. It has a method to add a card to the list. It also has a method that adds up the total value of the hand, taking into account the special behavior of ace (that it counts as one when it would cause the total to go above 33). It also overrides toString to return a string with a list of all the cards in the hand followed by its total value.

The driver class will have the user play a new game until he wins five games or quits. It uses a map with the key an enum of possible outcomes and the value an integer to store the outcomes of past games. For each game, it initializes the deck and the hands (one for the user (player) and one for the computer (dealer), picks three cards for the player, and offer to pick another card for the player until his hand equals 33. Then, it picks cards for the dealer until his hand exceeds 26. Whoever’s hand is closer to 33 without exceeding it wins. Following each game, it displays the number of previous games of each outcome and asks the user if he wants to play another game.