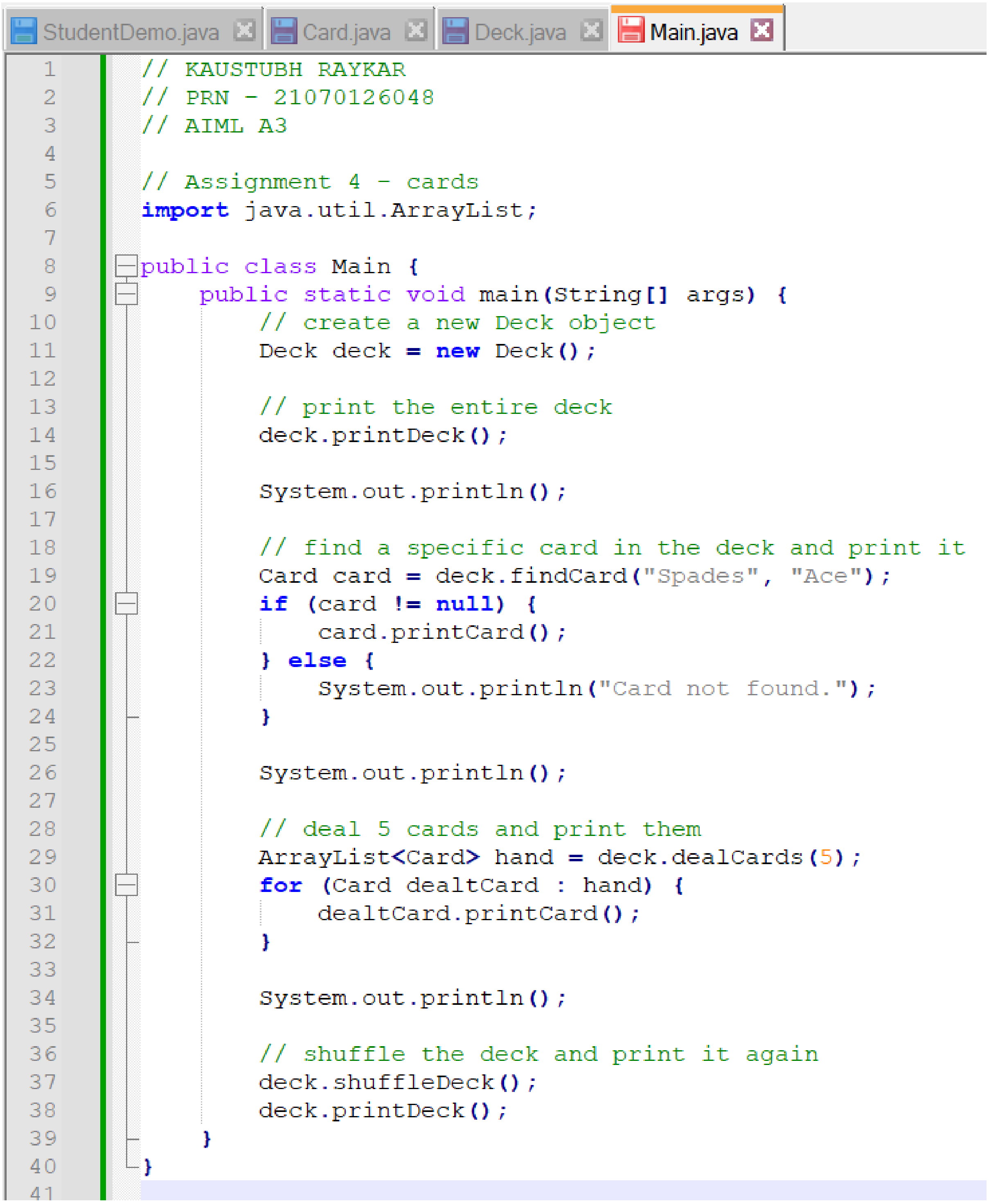
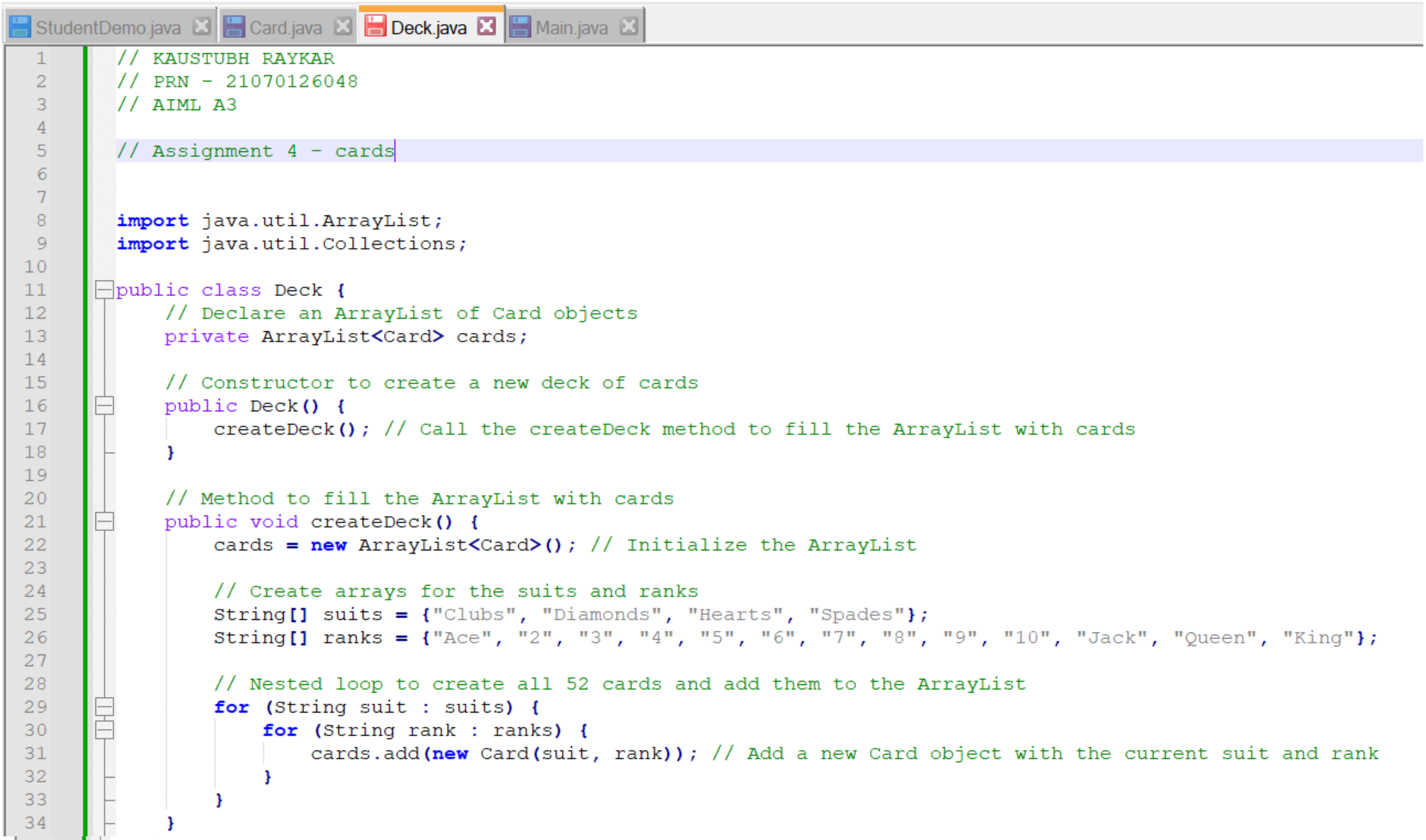
PIJ LAB Assignment 4

KAUSTUBH RAYKAR
PRN – 21070126048
AIML A3

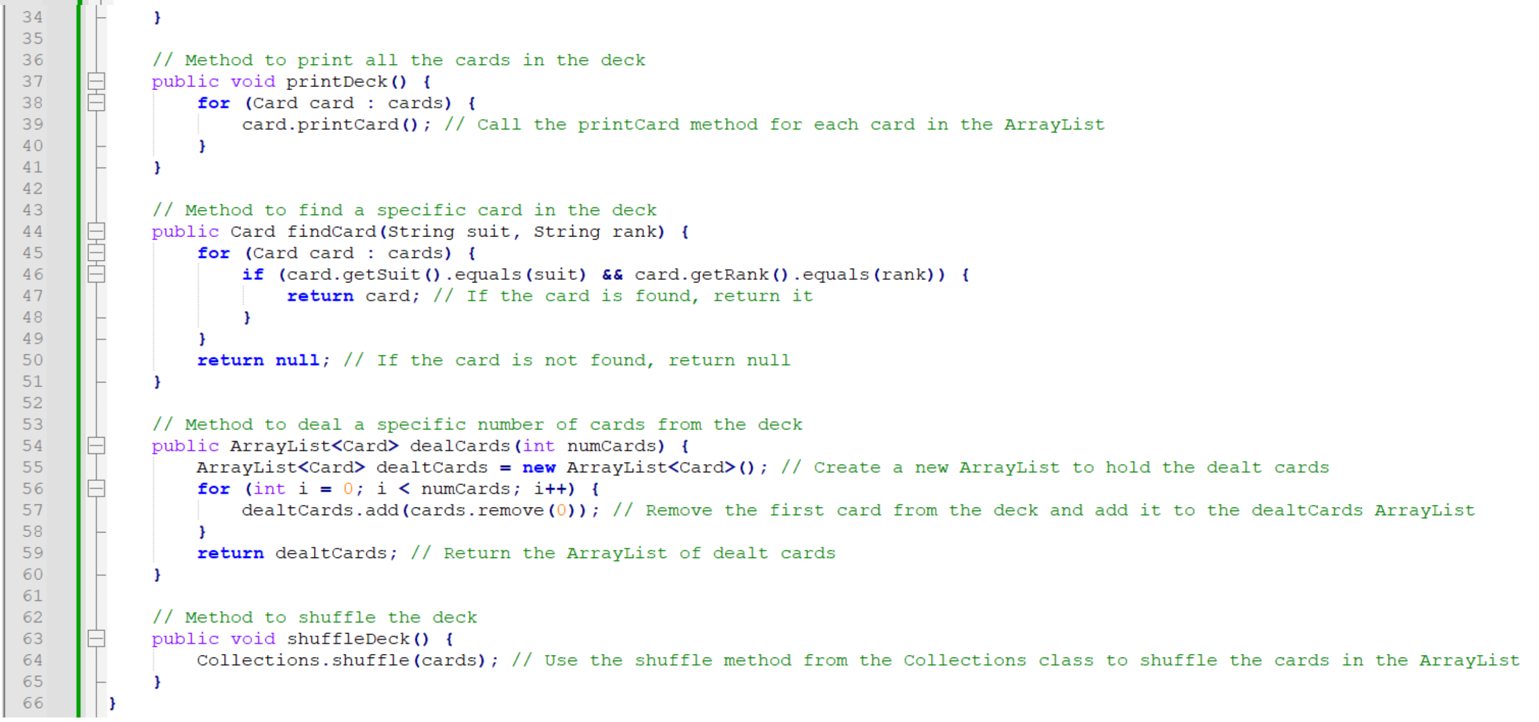
MAIN.JAVA



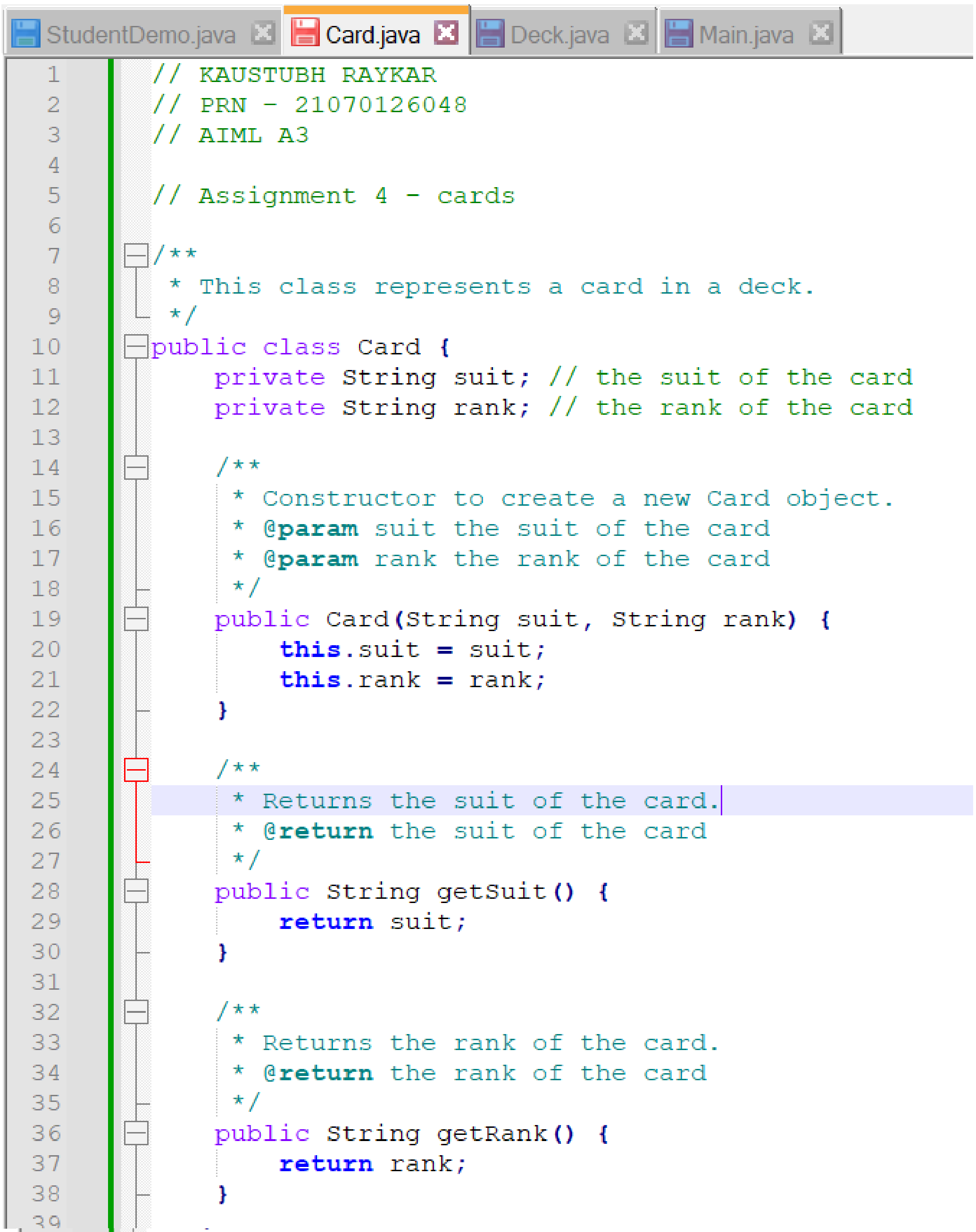
```java
// KAUSTUBH RAYKAR
// PRN - 21070126048
// AIML A3

// Assignment 4 - cards
import java.util.ArrayList;

public class Main {
    public static void main(String[] args) {
        // create a new Deck object
        Deck deck = new Deck();

        // print the entire deck
        deck.printDeck();

        System.out.println();

        // find a specific card in the deck and print it
        Card card = deck.findCard("Spades", "Ace");
        if (card != null) {
            card.printCard();
        } else {
            System.out.println("Card not found.");
        }

        System.out.println();

        // deal 5 cards and print them
        ArrayList<Card> hand = deck.dealCards(5);
        for (Card dealtCard : hand) {
            dealtCard.printCard();
        }

        System.out.println();

        // shuffle the deck and print it again
        deck.shuffleDeck();
        deck.printDeck();
    }
}
```



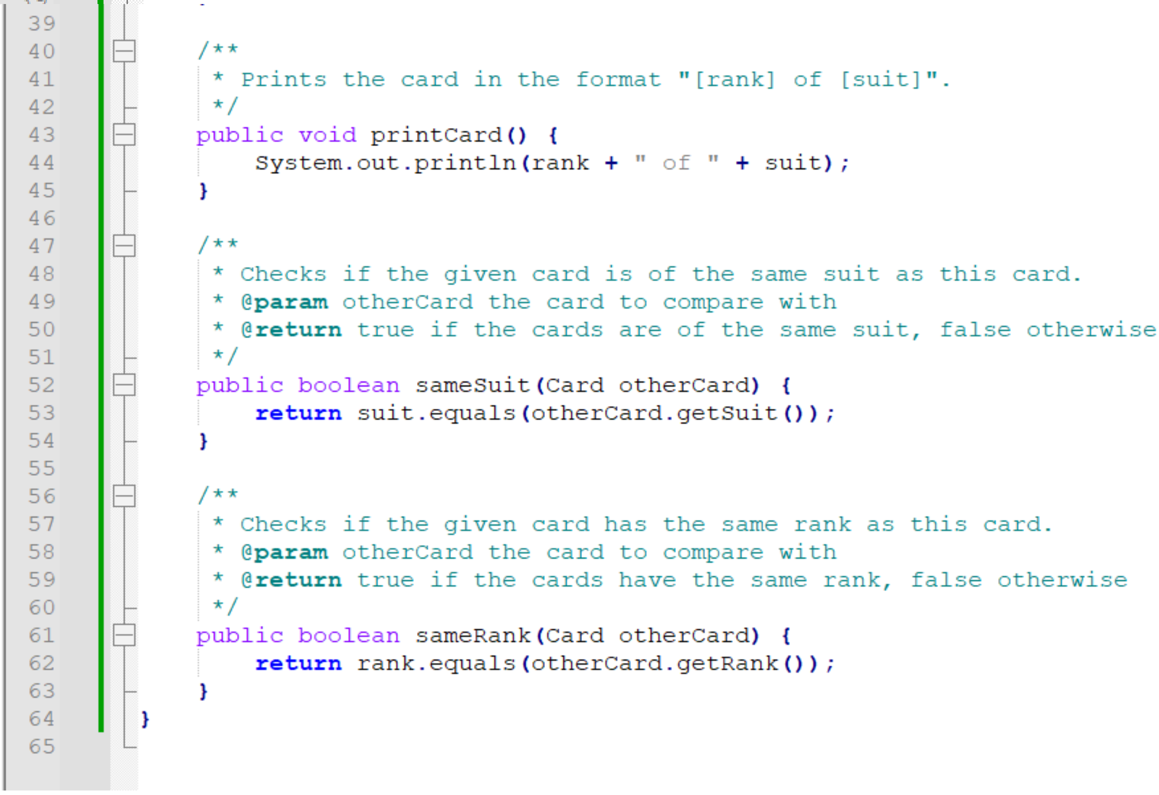
```java
// KAUSTUBH RAYKAR
// PRN - 21070126048
// AIML A3

// Assignment 4 - cards


import java.util.ArrayList;
import java.util.Collections;

public class Deck {
    // Declare an ArrayList of Card objects
    private ArrayList<Card> cards;

    // Constructor to create a new deck of cards
    public Deck() {
        createDeck(); // Call the createDeck method to fill the ArrayList with cards
    }

    // Method to fill the ArrayList with cards
    public void createDeck() {
        cards = new ArrayList<Card>(); // Initialize the ArrayList

        // Create arrays for the suits and ranks
        String[] suits = {"Clubs", "Diamonds", "Hearts", "Spades"};
        String[] ranks = {"Ace", "2", "3", "4", "5", "6", "7", "8", "9", "10", "Jack", "Queen", "King"};

        // Nested loop to create all 52 cards and add them to the ArrayList
        for (String suit : suits) {
            for (String rank : ranks) {
                cards.add(new Card(suit, rank)); // Add a new Card object with the current suit and rank
            }
        }
    }
```



```java
    }

    // Method to print all the cards in the deck
    public void printDeck() {
        for (Card card : cards) {
            card.printCard(); // Call the printCard method for each card in the ArrayList
        }
    }

    // Method to find a specific card in the deck
    public Card findCard(String suit, String rank) {
        for (Card card : cards) {
            if (card.getSuit().equals(suit) && card.getRank().equals(rank)) {
                return card; // If the card is found, return it
            }
        }
        return null; // If the card is not found, return null
    }

    // Method to deal a specific number of cards from the deck
    public ArrayList<Card> dealCards(int numCards) {
        ArrayList<Card> dealtCards = new ArrayList<Card>(); // Create a new ArrayList to hold the dealt cards
        for (int i = 0; i < numCards; i++) {
            dealtCards.add(cards.remove(0)); // Remove the first card from the deck and add it to the dealtCards ArrayList
        }
        return dealtCards; // Return the ArrayList of dealt cards
    }

    // Method to shuffle the deck
    public void shuffleDeck() {
        Collections.shuffle(cards); // Use the shuffle method from the Collections class to shuffle the cards in the ArrayList
    }
}
```



```java
// KAUSTUBH RAYKAR
// PRN - 21070126048
// AIML A3

// Assignment 4 - cards

/**
 * This class represents a card in a deck.
 */
public class Card {
    private String suit; // the suit of the card
    private String rank; // the rank of the card

    /**
     * Constructor to create a new Card object.
     * @param suit the suit of the card
     * @param rank the rank of the card
     */
    public Card(String suit, String rank) {
        this.suit = suit;
        this.rank = rank;
    }

    /**
     * Returns the suit of the card.
     * @return the suit of the card
     */
    public String getSuit() {
        return suit;
    }

    /**
     * Returns the rank of the card.
     * @return the rank of the card
     */
    public String getRank() {
        return rank;
    }
```



```java

    /**
     * Prints the card in the format "[rank] of [suit]".
     */
    public void printCard() {
        System.out.println(rank + " of " + suit);
    }

    /**
     * Checks if the given card is of the same suit as this card.
     * @param otherCard the card to compare with
     * @return true if the cards are of the same suit, false otherwise
     */
    public boolean sameSuit(Card otherCard) {
        return suit.equals(otherCard.getSuit());
    }

    /**
     * Checks if the given card has the same rank as this card.
     * @param otherCard the card to compare with
     * @return true if the cards have the same rank, false otherwise
     */
    public boolean sameRank(Card otherCard) {
        return rank.equals(otherCard.getRank());
    }
}

```