**PIJ Assignment 4**

Name: Samarth Bhadane

PRN: 24070126503

Batch: A2

**Code:**

// Card.java

public class Card {

    private String suit;

    private String rank;

    // Constructor to initialize card properties

    public Card(String suit, String rank) {

        this.suit = suit;

        this.rank = rank;

    }

    // Getters for Suit and Rank

    public String getSuit() {

        return suit;

    }

    public String getRank() {

        return rank;

    }

    // Print the card details

    public void printCard() {

        System.out.println(rank + " of " + suit);

    }

    // Check if two cards belong to the same suit

    public boolean sameCard(Card other) {

        return this.suit.equals(other.suit);

    }

    // Check if two cards have the same rank

    public boolean compareCard(Card other) {

        return this.rank.equals(other.rank);

    }

    // Check if a given card matches this card

    public boolean isEqual(Card other) {

        return this.suit.equals(other.suit) && this.rank.equals(other.rank);

    }

}

// Deck.java

import java.util.\*;

public class Deck {

    private ArrayList<Card> deck;

    private String[] suits = {"Hearts", "Diamonds", "Clubs", "Spades"};

    private String[] ranks = {"2", "3", "4", "5", "6", "7", "8", "9", "10",

                              "Jack", "Queen", "King", "Ace"};

    // Constructor to create a deck of 52 cards

    public Deck() {

        createDeck();

    }

    // Method to create deck

    public void createDeck() {

        deck = new ArrayList<>();

        for (String suit : suits) {

            for (String rank : ranks) {

                deck.add(new Card(suit, rank));

            }

        }

    }

    // Print all cards in the deck

    public void printDeck() {

        for (Card card : deck) {

            card.printCard();

        }

    }

    // Shuffle the deck randomly

    public void shuffleDeck() {

        Collections.shuffle(deck);

        System.out.println("Deck shuffled successfully!");

    }

    // Search for a specific card and return its index position

    public int findCard(String suit, String rank) {

        for (int i = 0; i < deck.size(); i++) {

            Card card = deck.get(i);

            if (card.getSuit().equalsIgnoreCase(suit) && card.getRank().equalsIgnoreCase(rank)) {

                System.out.println("Card found at index: " + i);

                return i;

            }

        }

    System.out.println("Card not found.");

    return -1;

}

    // Deal 5 random cards

    public void dealCard() {

        shuffleDeck();

        System.out.println("Dealing 5 random cards:");

        for (int i = 0; i < 5; i++) {

            deck.get(i).printCard();

        }

    }

}

// Main.java

import java.util.Scanner;

public class Main {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        Deck deck = new Deck();

        while (true) {

            System.out.println("\n--- CARD DECK MENU ---");

            System.out.println("1. Print Deck");

            System.out.println("2. Shuffle Deck");

            System.out.println("3. Search for a Card");

            System.out.println("4. Deal 5 Cards");

            System.out.println("5. Exit");

            System.out.print("Enter your choice: ");

            int choice = scanner.nextInt();

            scanner.nextLine(); // Consume newline

            switch (choice) {

                case 1:

                    deck.printDeck();

                    break;

                case 2:

                    deck.shuffleDeck();

                    break;

                case 3:

                    System.out.print("Enter card suit: ");

                    String suit = scanner.nextLine();

                    System.out.print("Enter card rank: ");

                    String rank = scanner.nextLine();

                    int position = deck.findCard(suit, rank);

                    if (position != -1) {

                        System.out.println("The card is located at index: " + position);

                    }

                    break;

                case 4:

                    deck.dealCard();

                    break;

                case 5:

                    System.out.println("Exiting program. Thank you!");

                    scanner.close();

                    return;

                default:

                    System.out.println("Invalid choice! Try again.");

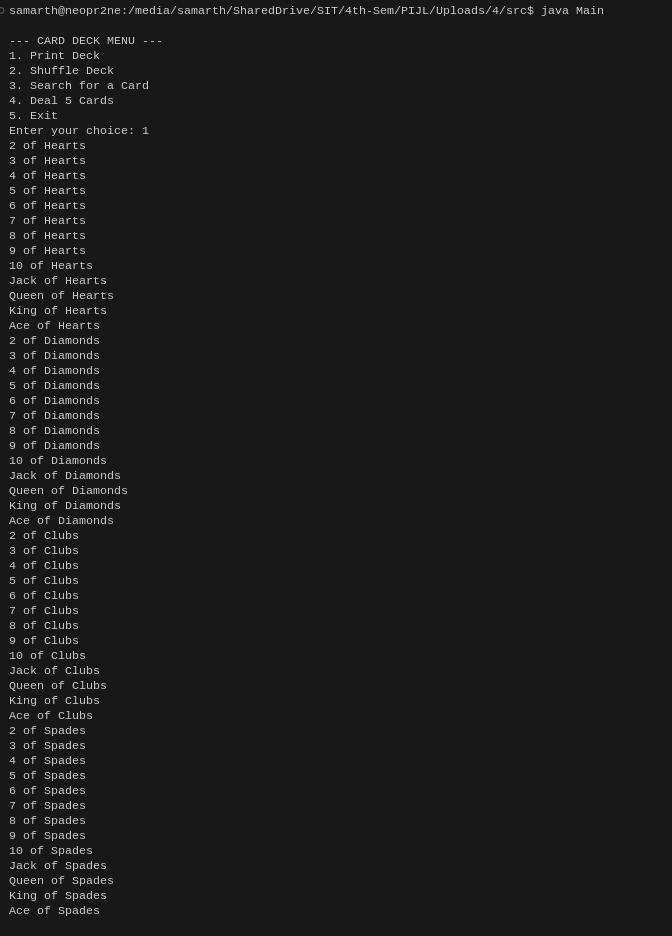
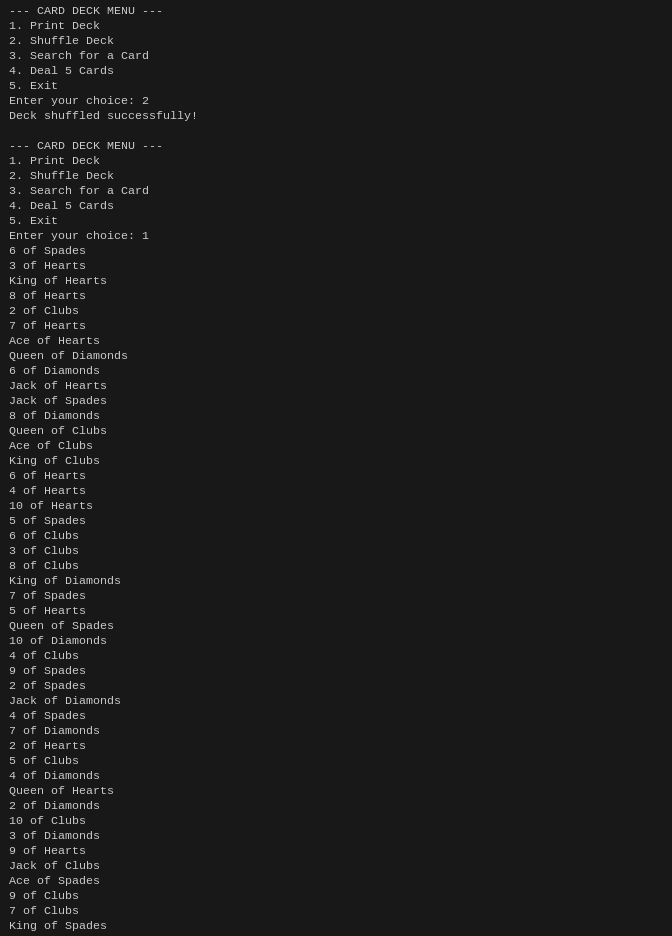
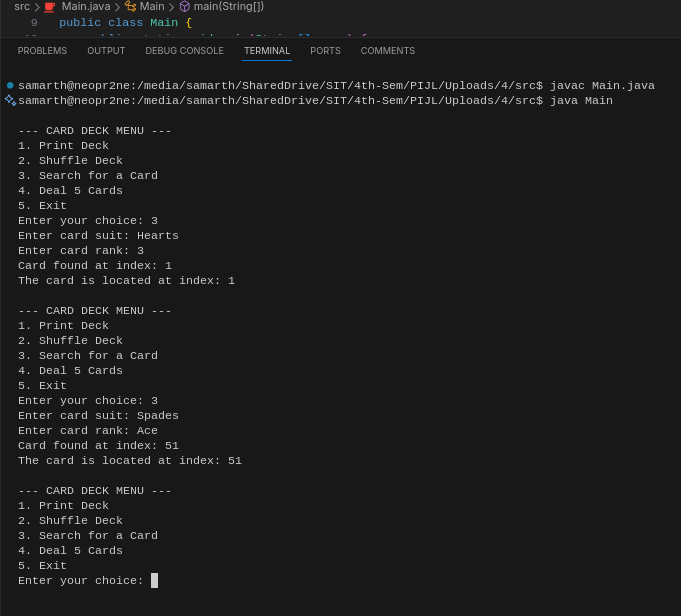
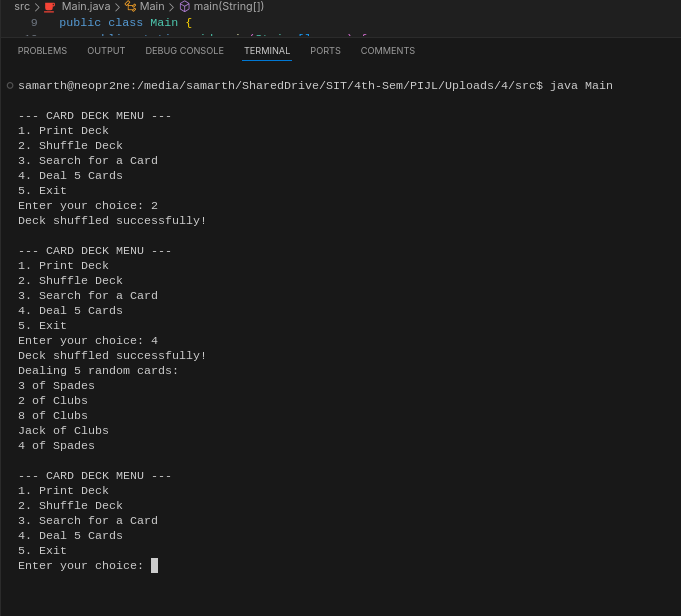
            }

        }

    }

}

**Output-**

1. Print Deck
2. Shuffle Deck
3. Search a Card
4. Find a Card

Link to the repository: <https://github.com/samarthsb4real/PIJ-Assignment-4.git>