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);
    }
}
```

```java
// 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();
        }
    }
}
```

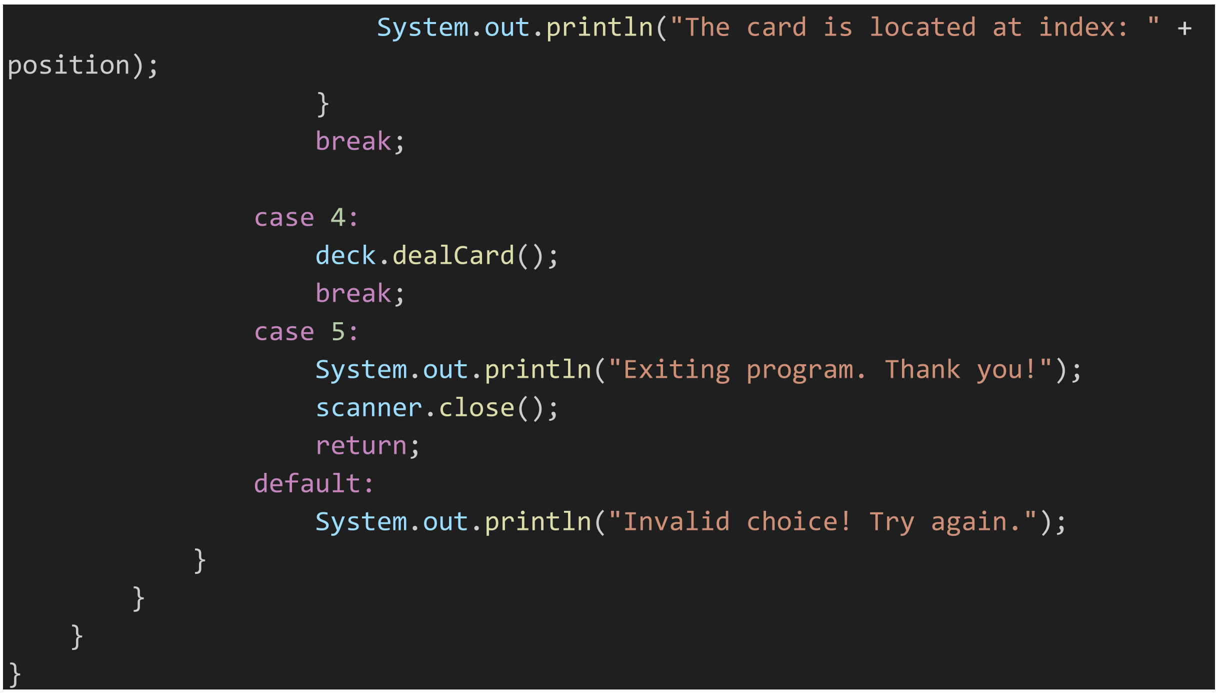
```
// Main.java
import java.util.Scanner;

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) {
```

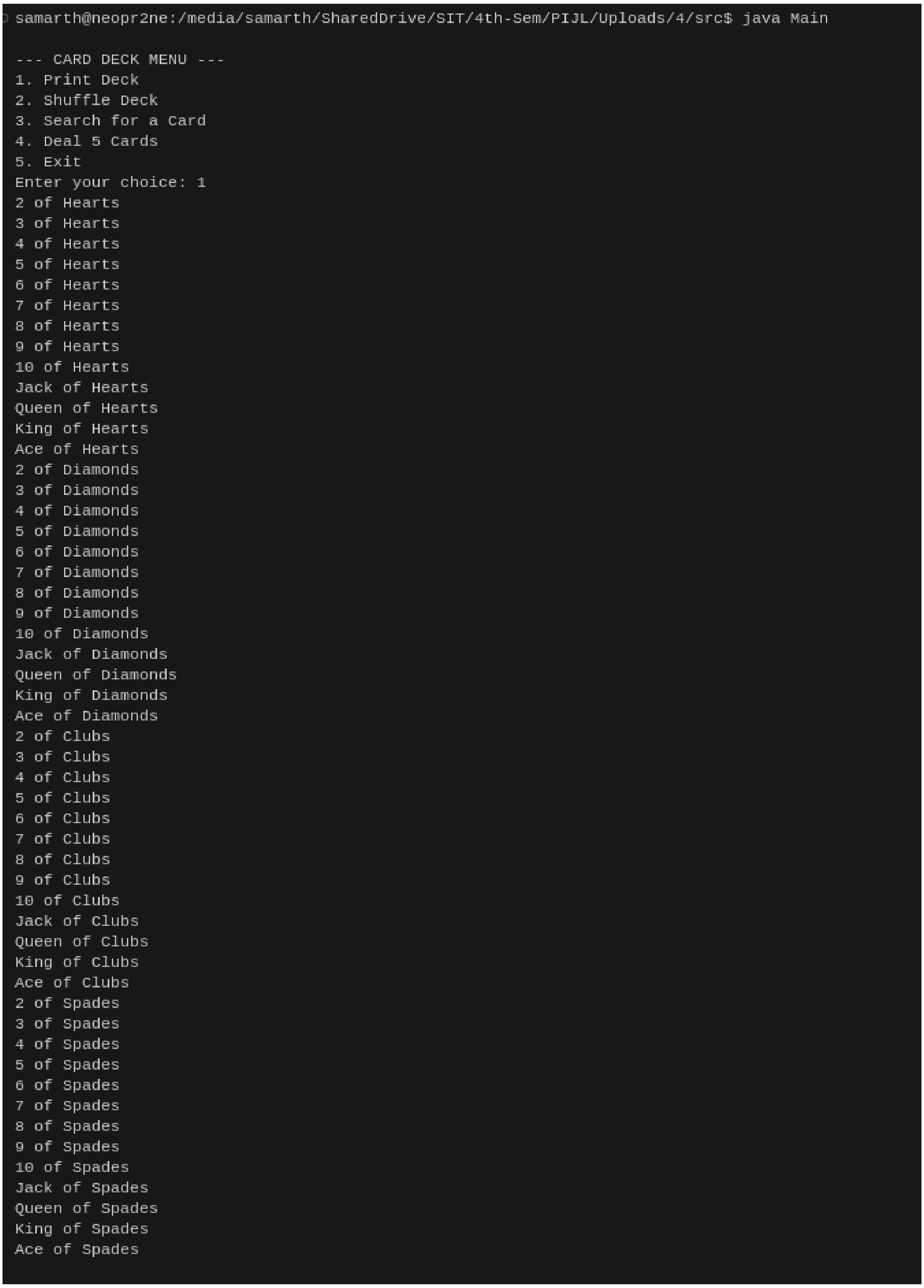


```
                        System.out.println("The card is located at index: " +
position);
                    }
                    break;

                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-

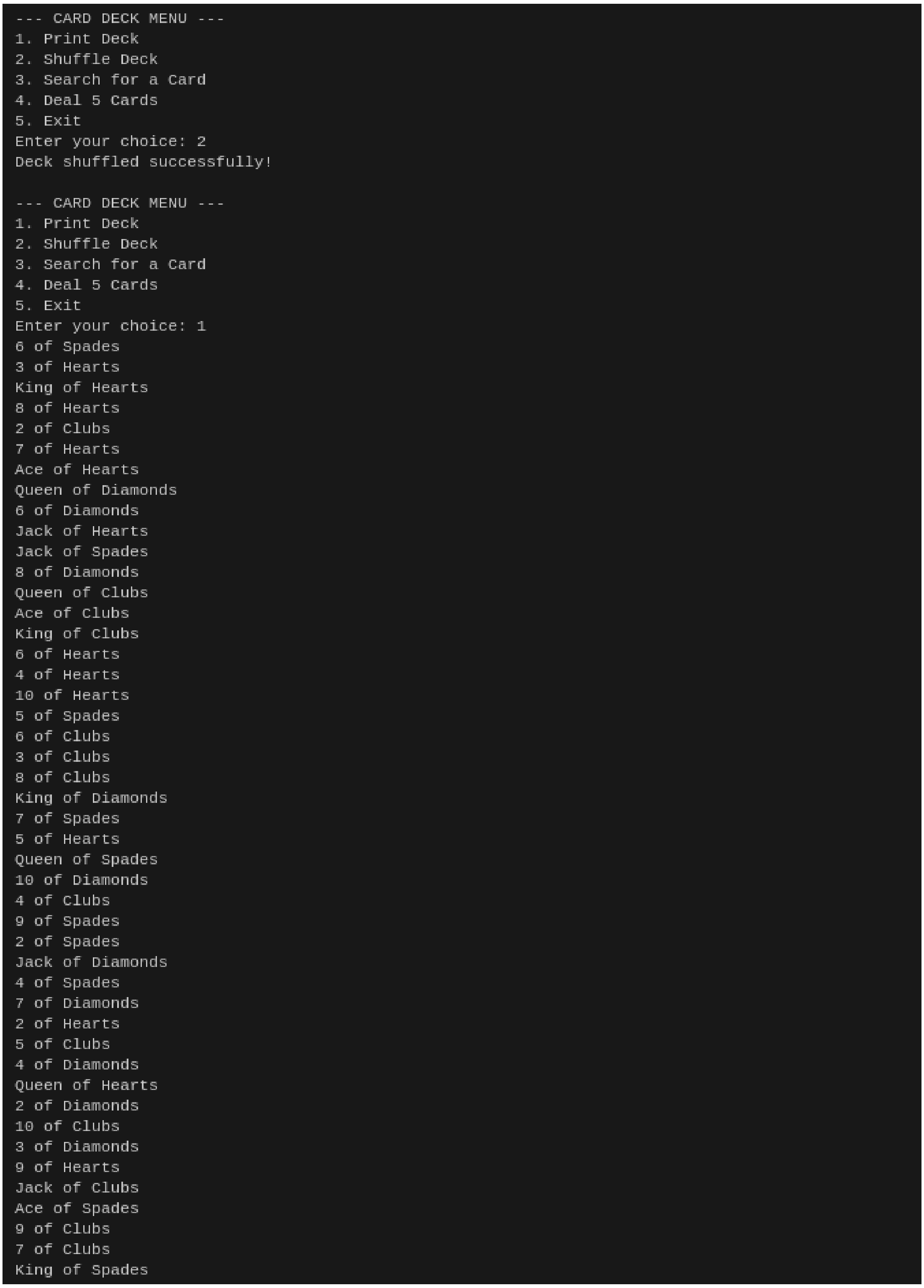
1. Print Deck



```
samarth@neopr2ne:/media/samarth/SharedDrive/SIT/4th-Sem/PIJL/Uploads/4/src$ java Main

--- CARD DECK MENU ---
1. Print Deck
2. Shuffle Deck
3. Search for a Card
4. Deal 5 Cards
5. Exit
Enter your choice: 1
2 of Hearts
3 of Hearts
4 of Hearts
5 of Hearts
6 of Hearts
7 of Hearts
8 of Hearts
9 of Hearts
10 of Hearts
Jack of Hearts
Queen of Hearts
King of Hearts
Ace of Hearts
2 of Diamonds
3 of Diamonds
4 of Diamonds
5 of Diamonds
6 of Diamonds
7 of Diamonds
8 of Diamonds
9 of Diamonds
10 of Diamonds
Jack of Diamonds
Queen of Diamonds
King of Diamonds
Ace of Diamonds
2 of Clubs
3 of Clubs
4 of Clubs
5 of Clubs
6 of Clubs
7 of Clubs
8 of Clubs
9 of Clubs
10 of Clubs
Jack of Clubs
Queen of Clubs
King of Clubs
Ace of Clubs
2 of Spades
3 of Spades
4 of Spades
5 of Spades
6 of Spades
7 of Spades
8 of Spades
9 of Spades
10 of Spades
Jack of Spades
Queen of Spades
King of Spades
Ace of Spades
```

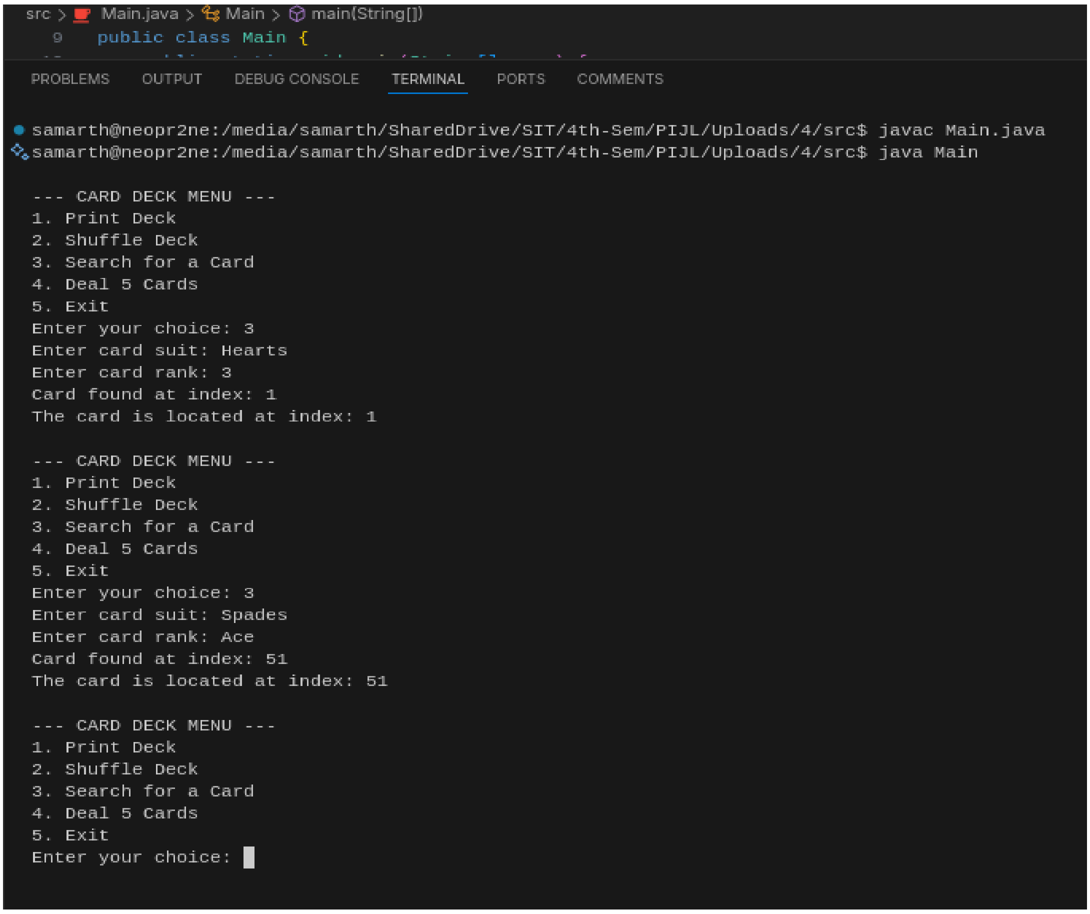
2. Shuffle Deck



```
--- CARD DECK MENU ---
1. Print Deck
2. Shuffle Deck
3. Search for a Card
4. Deal 5 Cards
5. Exit
Enter your choice: 2
Deck shuffled successfully!

--- CARD DECK MENU ---
1. Print Deck
2. Shuffle Deck
3. Search for a Card
4. Deal 5 Cards
5. Exit
Enter your choice: 1
6 of Spades
3 of Hearts
King of Hearts
8 of Hearts
2 of Clubs
7 of Hearts
Ace of Hearts
Queen of Diamonds
6 of Diamonds
Jack of Hearts
Jack of Spades
8 of Diamonds
Queen of Clubs
Ace of Clubs
King of Clubs
6 of Hearts
4 of Hearts
10 of Hearts
5 of Spades
6 of Clubs
3 of Clubs
8 of Clubs
King of Diamonds
7 of Spades
5 of Hearts
Queen of Spades
10 of Diamonds
4 of Clubs
9 of Spades
2 of Spades
Jack of Diamonds
4 of Spades
7 of Diamonds
2 of Hearts
5 of Clubs
4 of Diamonds
Queen of Hearts
2 of Diamonds
10 of Clubs
3 of Diamonds
9 of Hearts
Jack of Clubs
Ace of Spades
9 of Clubs
7 of Clubs
King of Spades
```

3. Search a Card



```java
public class Main {
```

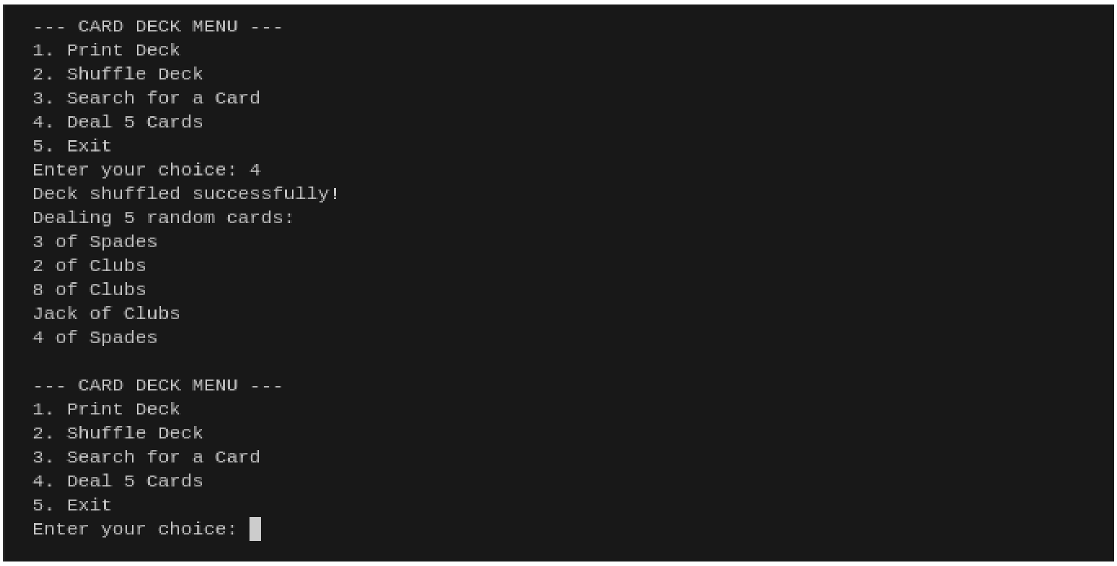
```
samarth@neopr2ne:/media/samarth/SharedDrive/SIT/4th-Sem/PIJL/Uploads/4/src$ javac Main.java
samarth@neopr2ne:/media/samarth/SharedDrive/SIT/4th-Sem/PIJL/Uploads/4/src$ java Main

 --- CARD DECK MENU ---
 1. Print Deck
 2. Shuffle Deck
 3. Search for a Card
 4. Deal 5 Cards
 5. Exit
 Enter your choice: 3
 Enter card suit: Hearts
 Enter card rank: 3
 Card found at index: 1
 The card is located at index: 1

 --- CARD DECK MENU ---
 1. Print Deck
 2. Shuffle Deck
 3. Search for a Card
 4. Deal 5 Cards
 5. Exit
 Enter your choice: 3
 Enter card suit: Spades
 Enter card rank: Ace
 Card found at index: 51
 The card is located at index: 51

 --- CARD DECK MENU ---
 1. Print Deck
 2. Shuffle Deck
 3. Search for a Card
 4. Deal 5 Cards
 5. Exit
 Enter your choice:
```

4. Find a Card



```
--- CARD DECK MENU ---
1. Print Deck
2. Shuffle Deck
3. Search for a Card
4. Deal 5 Cards
5. Exit
Enter your choice: 4
Deck shuffled successfully!
Dealing 5 random cards:
3 of Spades
2 of Clubs
8 of Clubs
Jack of Clubs
4 of Spades

--- CARD DECK MENU ---
1. Print Deck
2. Shuffle Deck
3. Search for a Card
4. Deal 5 Cards
5. Exit
Enter your choice:
```

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