Name: Aadith Sukumar
Branch: AIML A1
Batch: 2021-2025
PRN: 21070126003

Programming In Java
ASSIGNMENT 4

Question:

Write a menu-driven Java Program for the following:
There are 52 cards in a deck, each of which belongs to one of four suits and one of 13 ranks. Represent a deck of cards as an array of Objects (*you may use the Vector class)

1. Use integers to encode the ranks and suits.
2. Have suitable default & parameterized constructors.
3. all data members to have private access.
4. The class 'Card' to have the following methods:
   createDeck()
   printCard()
   printDeck ()
   sameCard()
   compareCard()
   sortCard()
   findCard() which searches through an array or vector of Cards to see whether it contains a certain card

Code:

Card.Java

```
/*  A menu-driven Java Program for the following: There are 52 cards in
*   a deck, each of which belongs to one of four suits and one of 13 ranks.
*   Represent a deck of cards as an array of Objects
*/
// Code by Aadith Sukumar (https://github.com/aadi1011)

public class Card
{
    private String suit;
    private String value;

    public Card(String suit, String value)
    {
        this.suit = suit;
        this.value = value;
    }
```

```
    public String getSuit()
    {
        return suit;
    }

    public void setSuit(String suit)
    {
        this.suit = suit;
    }

    public String getValue()
    {
        return value;
    }

    public void setValue(String value)
    {
        this.value = value;
    }

    public String toString()
    {
        return value + " of " + suit;
    }

}
```

Deck.Java

```
/* Operation function codes for Card.java
 * Code by Aadith Sukumar
 */

package CardsClasses;

import java.util.ArrayList;
import java.util.Scanner;

public class Deck
{
    ArrayList<Card> card = new ArrayList<Card>();
    String[] suits = {"Hearts", "Diamonds", "Spades", "Clubs"};
    String[] values = {"Ace", "2", "3", "4", "5", "6", "7", "8", "9", "10",
"Jack", "Queen", "King"};

    // Constructor - creates a deck of cards and adds them to the ArrayList
    public Deck() {
        for (int i = 0; i < suits.length; i++) {
            for (int j = 0; j < values.length; j++) {
```

```
                card.add(new Card(suits[i], values[j]));
            }
        }
    }

    // returns the number of cards in the deck
    public int size() {
        return card.size();
    }

    // print the entire deck
    public void printDeck()
    {

        for (int i = 0; i < card.size(); i++)
        {
            System.out.println(card.get(i));
        }
        menu();
    }

    // print one card randomly
    public void printCard()
    {
        int random = (int) (Math.random() * card.size());
        System.out.println("\n========== Output: Random Card ==========\n");
        System.out.println("Card Drawn: "+card.get(random));
        menu();
    }

    // deck shuffling
    public void shuffle()
    {
        for (int i = 0; i < card.size(); i++)
        {
            int random = (int) (Math.random() * card.size());
            Card temp = card.get(i);
            card.set(i, card.get(random));
            card.set(random, temp);
        }
        System.out.println("\n========== Output: Deck Shuffled ==========\n");
        menu();
    }

    // function to delete a specific card from the deck
    public void deleteCard()
    {
```

```
        System.out.println("Enter the suit of the card you want to delete
['Hearts', 'Diamonds', 'Spades', 'Clubs']:  ");
        Scanner sc = new Scanner(System.in);
        String suit = sc.nextLine();
        System.out.println("Enter the value of the card you want to delete
['Ace', '2', '3', '4', '5', '6', '7', '8', '9', '10', 'Jack', 'Queen',
'King']: ");
        String value = sc.nextLine();
        int removecounter=0;
        for (int i = 0; i < card.size(); i++)
        {
            if (card.get(i).getSuit().equals(suit) &&
card.get(i).getValue().equals(value))
            {
                card.remove(i);
                removecounter=1;
                System.out.println("\nCard removed");
            }
            menu();
        }
        if(removecounter==0)
            {
                System.out.println("\nCard not found");
            }
        sc.close();
    }

    //function to delete a suit of cards from the deck
    public void deleteDeck()
    {
        System.out.println("Enter the suit of the card you want to delete
['Hearts', 'Diamonds', 'Spades', 'Clubs']:  ");
        Scanner sc = new Scanner(System.in);
        String suit = sc.nextLine();
        int removecounter=0;
        for (int i = 0; i < card.size(); i++)
        {
            if (card.get(i).getSuit().equals(suit))
            {
                card.remove(i);
                removecounter=1;
                System.out.println("\nAll cards of " + suit + "removed");
            }
        }
        if(removecounter==0)
            {
                System.out.println("\nNo cards of " + suit + "found");
            }
```

```
        menu();
        sc.close();
    }

    // function to add a card to the deck
    public void addCard()
    {
        System.out.println("\nEnter the suit of the card you want to add:
['Hearts', 'Diamonds', 'Spades', 'Clubs']:   ");
        Scanner sc = new Scanner(System.in);
        String suit = sc.nextLine();
        System.out.println("\nEnter the value of the card you want to add: ");
        String value = sc.nextLine();
        card.add(new Card(suit, value));
        System.out.println("Card added");
        menu();
        sc.close();
    }

    // function to add a deck of cards to the deck if any of the values of the
suit are missing
    public void addDeck()
    {
        System.out.println("Enter the suit of the card you want to add:
['Hearts', 'Diamonds', 'Spades', 'Clubs']:   ");
        Scanner sc = new Scanner(System.in);
        String suit = sc.nextLine();

        for (int i = 0; i < values.length; i++)
        {
            card.add(new Card(suit, values[i]));
        }
        System.out.println("Deck added");
        menu();
        sc.close();
    }

    // recursive menu function
    public void menu()
    {
        System.out.println("\n\n========== MENU ==========");

        System.out.println("\n1. Print the deck");
        System.out.println("2. Print a random card");
        System.out.println("3. Shuffle the deck");
        System.out.println("4. Delete a card");
        System.out.println("5. Add a card");
        System.out.println("6. Delete a deck");
```

```
        System.out.println("7. Exit");
        System.out.println("Enter your choice: ");
        Scanner sc = new Scanner(System.in);
        int choice = sc.nextInt();

        switch (choice)
        {
            case 1:
                printDeck();
                break;
            case 2:
                printCard();
                break;
            case 3:
                shuffle();
                break;
            case 4:
                deleteCard();
                break;
            case 5:
                addCard();
                break;
            case 6:
                deleteDeck();
                break;
            case 7:
                System.exit(0);
                break;
            default:
                System.out.println("Invalid choice");
                break;
        }
        sc.close();
    }


    public static void main(String[] args)
    {
        Deck deck = new Deck();
        deck.menu();
    }
}
```

Output:

```
========== MENU ==========

1. Print the deck
2. Print a random card
3. Shuffle the deck
4. Delete a card
5. Add a card
6. Delete a deck
7. Exit
Enter your choice:
2

========== Output: Random Card ==========

Card Drawn: Queen of Diamonds
```

```
========== MENU ==========

1. Print the deck
2. Print a random card
3. Shuffle the deck
4. Delete a card
5. Add a card
6. Delete a deck
7. Exit
Enter your choice:
3

========== Output: Deck Shuffled ==========



========== MENU ==========

1. Print the deck
2. Print a random card
3. Shuffle the deck
4. Delete a card
5. Add a card
6. Delete a deck
7. Exit
Enter your choice:
1
10 of Hearts
8 of Diamonds
9 of Hearts
3 of Clubs
5 of Hearts
2 of Clubs
King of Hearts
5 of Clubs
Ace of Hearts
Jack of Clubs
6 of Spades
10 of Spades
3 of Diamonds
Jack of Hearts
9 of Diamonds
```

GitHub Repo Link:

https://github.com/aadi1011/Basic-Java-Programs/tree/main/Assignment%204