# Java lab Assignment 4

Om Varshney. AI ML B2. 21070126117.

## Card.java

package Assn\_4;

public class Card {

    private int type, value;

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

    private String[] cardValue = {"Ace", "King", "Queen", "Jack", "10","9", "8", "7", "6", "5", "4", "3", "2"};

    public Card(int type, int value) {

        this.type = type;

        this.value = value;

    }

    public String toString() {

        String finalCard = cardValue[value] + " of " + cardType[type];

        return finalCard;

    }

    public static boolean equalTo(Card firstCard, Card secondCard) {

        return firstCard.type == secondCard.type && firstCard.value == secondCard.value;

    }

    public static boolean greaterThan(Card firstCard, Card secondCard) {

        return firstCard.type > secondCard.type || firstCard.value > secondCard.value;

    }

}

## Deck.java

package Assn\_4;

import java.util.Collections;

import java.util.Random;

import java.util.Vector;

public class Deck {

    private Vector<Card> cards;

    public Deck() {

        cards = new Vector<Card>();

        for(int a=0; a<=3; a++) {

            for(int b=0; b<=12; b++) {

                cards.add(new Card(a,b));

            }

        }

    }

    public Card drawRandomCard() {

        Random generator = new Random();

        int index = generator.nextInt(cards.size());

        return cards.remove(index);

    }

    public boolean sameCard(Card firstCard, Card secondCard) {

        // Returns true if cards are equal. false otherwise.

        return Card.equalTo(firstCard, secondCard);

    }

    public static boolean compareCard(Card firstCard, Card secondCard) {

        // Returns true if first card is greater than second card. false otherwise.

        return Card.greaterThan(firstCard, secondCard);

    }

    public void sortDeck() {

        // A simple code to use bubble sort to swap the cards.

        int maxLimit = cards.size();

        for (int i = 0; i < maxLimit; i++) {

            for (int j = 1; j < maxLimit - i - 1; j++) {

                if (compareCard(cards.get(j), cards.get(j + 1))) {

                    Collections.swap(cards, j, j + 1);

                }

            }

        }

    }

    public boolean findCard(Card card) {

        return cards.contains(card);

    }

    public void dealCards() {

        // Deal 5 random cards from the deck

        for (int i = 0; i < 5; i++) {

            int maxLimit = cards.size();

            int randIdx = (int) (Math.random() \* (maxLimit + 1));

            System.out.println(cards.remove(randIdx));

        }

    }

    public void printDeck() {

        for (Card card : cards) {

            System.out.println(card);

        }

    }

    public String toString() {

        String result = "Cards remaining in deck: " + cards;

        return result;

    }

}

## Main.java

package Assn\_4;

import java.util.Scanner;

/\*

In this Java Program we were supposed to simulate a deck of cards using java vector. We have built the following functions-

1. printDeck

2. printCard

3. sortDeck

4. dealCards

5. drawRandomCard

and others which complete the implementation of deck of cards in java.

\*/

public class Main {

    public static void main(String[] args) {

        Scanner scan = new Scanner(System.in);

        Card C;

        Deck deck = new Deck();

        deck.printDeck();

        System.out.print("Enter number of cards to be dealt: ");

        int numberCards = scan.nextInt();

        System.out.println("\nCards drawn: \n");

        for (int i = 0; i < numberCards; i++) {

            C = deck.drawRandomCard();

            System.out.println(C);

        }

        scan.close();

        System.out.println("\nDealing Random Cards\n");

        deck.dealCards();

        System.out.println("\nSorting the deck\n");

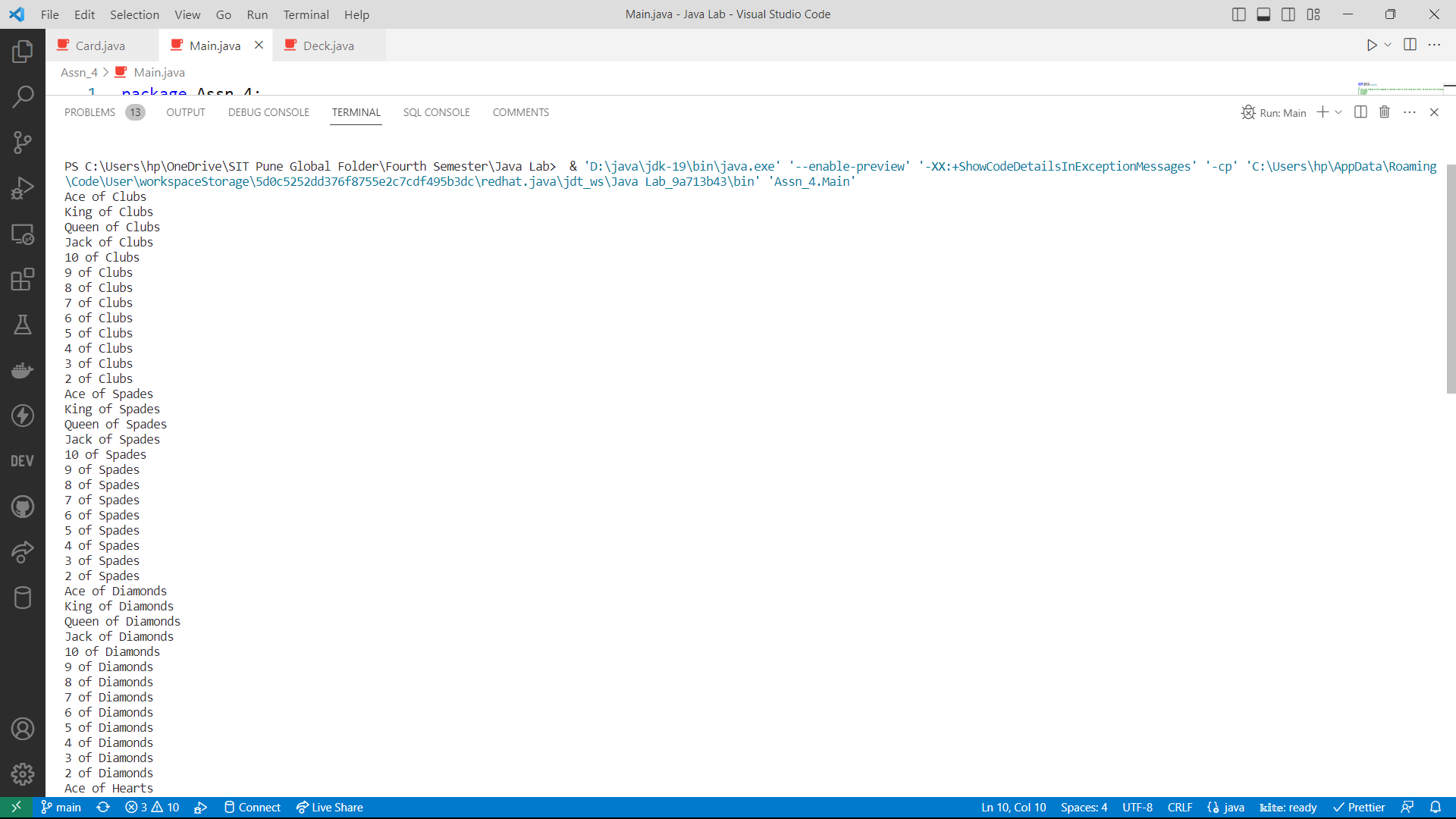
        deck.sortDeck();

        deck.printDeck();

    }

}

## Outputs



Graphical user interface, text, application, Word

Description automatically generated

Graphical user interface, text, application, Word

Description automatically generated

## GitHub Repository Link

https://github.com/om-varshney/Java-Lab/tree/main/Assn\_4