```java
//Name : Abhishek B
//PRN: 21070126007

// Problem: 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.

import java.util.*;

public class Assignment_vector {
    public static void main(String[] args) {
        Deck deck = new Deck();
        deck.createDeck();

    }
}

class Card {
    public static final String[] suits = { "Hearts", "Diamonds", "Clubs", "Spades" };
    public static final String[] ranks = { "Ace", "2", "3", "4", "5", "6", "7", "8", "9", "10", "Jack", "Queen",
            "King" };

    private int rank;
    private String suit;

    public Card(int rank, String suit) {
        this.rank = rank;
        this.suit = suit;
    }

    public int getRank() {
        return rank;
    }

    public String getSuit() {
        return suit;
    }

    public String toString() {
        return ranks[rank - 1] + " of " + suit;
    }
}

// deck class
class Deck {
```

```
45      public void createDeck() {
46          Scanner input = new Scanner(System.in);
47          Vector<Card> deck = new Vector<Card>(52);
48
49          // populate the deck with cards
50          for (int rank = 1; rank <= 13; rank++) {
51              for (String suit : Card.suits) {
52                  Card card = new Card(rank, suit);
53                  deck.add(card);
54              }
55          }
56
57          // display the menu
58          while (true) {
59              System.out.println("**************************\n"
    );
60              System.out.println("1. Display the deck of cards");
61              System.out.println("2. Shuffle the deck of cards");
62              System.out.println("3. Draw a card from the deck");
63              System.out.println("4. Empty the deck");
64              System.out.println("5. Print a card from the deck"
    );
65              System.out.println("6. Compare two cards");
66              System.out.println("7. Check if two cards are same"
    );
67              System.out.println("8. Find card by rank and suit"
    );
68              System.out.println("9. Deal a hand of cards");
69              System.out.println("10. Quit");
70              // get user choice
71              System.out.print("Enter your choice (1-10): ");
72              int choice = input.nextInt();
73
74              System.out.println("**************************\n"
    );
75
76              // handle user choice
77              switch (choice) {
78                  case 1:
79                      displayDeck(deck);
80                      break;
81                  case 2:
82                      shuffleDeck(deck);
83                      break;
84                  case 3:
85                      drawCard(deck);
86                      break;
87                  case 4:
```

```
 88                      emptyDeck(deck);
 89                      break;
 90                  case 5:
 91                      printCard(deck);
 92                      break;
 93                  case 6:
 94                      compareCard(deck);
 95                      break;
 96                  case 7:
 97                      sameCard(deck);
 98                      break;
 99                  case 8:
100                      findCard(deck);
101                      break;
102                  case 9:
103                      dealCard(deck);
104                      break;
105                  case 10 :
106                      System.out.println("Goodbye!");
107                      System.exit(0);
108                  default:
109                      System.out.println("Invalid choice. Please
    try again.");
110                      break;
111              }
112          }
113      }
114
115      // display the current state of the deck
116      public static void displayDeck(Vector<Card> deck) {
117          System.out.println("Deck of Cards:");
118          for (Card card : deck) {
119              System.out.println(card);
120          }
121          System.out.println();
122      }
123
124      // shuffle the deck
125      public static void shuffleDeck(Vector<Card> deck) {
126          Collections.shuffle(deck);
127          System.out.println("Deck shuffled.");
128      }
129
130      // draw a card from the deck
131      public static void drawCard(Vector<Card> deck) {
132          if (deck.isEmpty()) {
133              System.out.println("Deck is empty.");
134          } else {
```

```
135             Card card = deck.remove(0);
136             System.out.println("You drew: " + card);
137         }
138     }
139 
140     // empty the deck
141     public static void emptyDeck(Vector<Card> deck) {
142         deck.clear();
143         System.out.println("Deck emptied.");
144     }
145 
146     // printCard() function take the input position in the
    deck and print the card
147     public static void printCard(Vector<Card> deck) {
148         Scanner input = new Scanner(System.in);
149         System.out.print("Enter the position of the card you
    want to draw: ");
150         int position = input.nextInt();
151 
152         if (deck.isEmpty()) {
153             System.out.println("Deck is empty.");
154         } else {
155             Card card = deck.get(position);
156             System.out.println("You drew: " + card);
157         }
158     }
159 
160     // sameCard() draws 2 random cards and compare their ranks
     to check if they are
161     // same or not
162     public static void sameCard(Vector<Card> deck) {
163         Random rand = new Random();
164         int firstCard = rand.nextInt(52);
165         int secondCard = rand.nextInt(52);
166 
167         if (deck.isEmpty()) {
168             System.out.println("Deck is empty.");
169         } else {
170             Card card1 = deck.get(firstCard);
171             Card card2 = deck.get(secondCard);
172             if (card1.getRank() == card2.getRank()) {
173                 System.out.println("You drew: " + card1 + "
    and " + card2 + " and they are ranked same.");
174             } else {
175                 System.out.println("You drew: " + card1 + "
    and " + card2 + " and they are not ranked same.");
176             }
177         }
```

```
178     }
179 
180     // compareCard() draws 2 random cards and compare them to
    get the card of higher
181     // rank and if ranks are same then compare their suits.
182     public static void compareCard(Vector<Card> deck) {
183         Random rand = new Random();
184         int firstCard = rand.nextInt(52);
185         int secondCard = rand.nextInt(52);
186 
187         if (deck.isEmpty()) {
188             System.out.println("Deck is empty.");
189         }
190         else
191         {
192             Card card1 = deck.get(firstCard);
193             Card card2 = deck.get(secondCard);
194             if (card1.getRank() > card2.getRank()) {
195                 System.out.println("You drew: " + card1 + "
    and " + card2 + " and " + card1 + " is of higher rank.");
196             } else if (card1.getRank() < card2.getRank()) {
197                 System.out.println("You drew: " + card1 + "
    and " + card2 + " and " + card2 + " is of higher rank.");
198             } else {
199                 if (card1.getSuit().equals("Hearts")) {
200                     System.out
201                             .println("You drew: " + card1 + "
    and " + card2 + " and " + card1 + " is of higher rank.");
202                 } else if (card2.getSuit().equals("Hearts")) {
203                     System.out
204                             .println("You drew: " + card1 + "
    and " + card2 + " and " + card2 + " is of higher rank.");
205                 } else if (card1.getSuit().equals("Diamonds"
    )) {
206                     System.out
207                             .println("You drew: " + card1 + "
    and " + card2 + " and " + card1 + " is of higher rank.");
208                 } else if (card2.getSuit().equals("Diamonds"
    )) {
209                     System.out
210                             .println("You drew: " + card1 + "
    and " + card2 + " and " + card2 + " is of higher rank.");
211                 } else if (card1.getSuit().equals("Clubs")) {
212                     System.out
213                             .println("You drew: " + card1 + "
    and " + card2 + " and " + card1 + " is of higher rank.");
214                 } else if (card2.getSuit().equals("Clubs")) {
215                     System.out
```

```
216                             .println("You drew: " + card1 + "
    and " + card2 + " and " + card2 + " is of higher rank.");
217                 }
218             }
219         }
220     }
221 
222     // sortCard() function sorts the deck of cards in
    ascending order of rank and if
223     // ranks are same then sort them in ascending order of
    suits.
224     public static void sortDeck(Vector<Card> deck) {
225         Collections.sort(deck, new Comparator<Card>() {
226             @Override
227             public int compare(Card card1, Card card2) {
228                 if (card1.getRank() == card2.getRank()) {
229                     return card1.getSuit().compareTo(card2.
    getSuit());
230                 } else {
231                     return card1.getRank() - card2.getRank();
232                 }
233             }
234         });
235         System.out.println("Deck of Cards:");
236         for (Card card : deck) {
237             System.out.println(card);
238         }
239         System.out.println();
240     }
241 
242     // findCard() function takes the input rank and suit and
    search the deck of cards to find the card with the given rank
    and suit. returns position of the card in the deck.
243     public static void findCard(Vector<Card> deck) {
244         Scanner input = new Scanner(System.in);
245         System.out.print("Enter the rank of the card you want
    to find: ");
246         int rank = input.nextInt();
247         System.out.print("Enter the suit (\"Hearts\", \"
    Diamonds\", \"Clubs\", \"Spades\") of the card you want to
    find: ");
248         String suit = input.next();
249 
250         if (deck.isEmpty()) {
251             System.out.println("Deck is empty.");
252         } else {
253             for (int i = 0; i < deck.size(); i++) {
254                 Card card = deck.get(i);
```

```
255                 if (card.getRank() == rank && card.getSuit().
    equals(suit)) {
256                     System.out.println("Card found at position
     " + i + " in the deck.");
257                     break;
258                 }
259             }
260         }
261     }
262 
263     // dealCard() function takes the input number of players
    and deal the cards to the players.
264     public static void dealCard(Vector<Card> deck) {
265         shuffleDeck(deck);
266         Scanner input = new Scanner(System.in);
267         System.out.print("Enter the number of players: ");
268         int players = input.nextInt();
269 
270         if (deck.isEmpty()) {
271             System.out.println("Deck is empty.");
272         } else {
273             int cardsPerPlayer = deck.size() / players;
274             int remainingCards = deck.size() % players;
275             int start = 0;
276             int end = cardsPerPlayer;
277             for (int i = 0; i < players; i++) {
278                 System.out.println("\nPlayer " + (i + 1) + "
    cards:");
279                 for (int j = start; j < end; j++) {
280                     System.out.println(deck.get(j));
281                 }
282                 start = end;
283                 end += cardsPerPlayer;
284             }
285             if (remainingCards > 0) {
286                 System.out.println("Remaining cards:");
287                 for (int i = end; i < deck.size(); i++) {
288                     System.out.println(deck.get(i));
289                 }
290             }
291         }
292     }
293 }
```

```
 1 "C:\Program Files\Java\jdk-19\bin\java.exe" "-
   javaagent:C:\Program Files\JetBrains\IntelliJ IDEA
   Community Edition 2022.3.1\lib\idea_rt.jar=51748:C:\
   Program Files\JetBrains\IntelliJ IDEA Community
   Edition 2022.3.1\bin" -Dfile.encoding=UTF-8 -Dsun.
   stdout.encoding=UTF-8 -Dsun.stderr.encoding=UTF-8 -
   classpath "D:\College\Fourth SEM\java\javA\out\
   production\javA" Assignment_vector
 2 ***************************
 3
 4 1. Display the deck of cards
 5 2. Shuffle the deck of cards
 6 3. Draw a card from the deck
 7 4. Empty the deck
 8 5. Print a card from the deck
 9 6. Compare two cards
10 7. Check if two cards are same
11 8. Find card by rank and suit
12 9. Deal a hand of cards
13 10. Quit
14 Enter your choice (1-10): 1
15 ***************************
16
17 Deck of Cards:
18 Ace of Hearts
19 Ace of Diamonds
20 Ace of Clubs
21 Ace of Spades
22 2 of Hearts
23 2 of Diamonds
24 2 of Clubs
25 2 of Spades
26 3 of Hearts
27 3 of Diamonds
28 3 of Clubs
29 3 of Spades
30 4 of Hearts
31 4 of Diamonds
32 4 of Clubs
33 4 of Spades
34 5 of Hearts
```

```
35 5 of Diamonds
36 5 of Clubs
37 5 of Spades
38 6 of Hearts
39 6 of Diamonds
40 6 of Clubs
41 6 of Spades
42 7 of Hearts
43 7 of Diamonds
44 7 of Clubs
45 7 of Spades
46 8 of Hearts
47 8 of Diamonds
48 8 of Clubs
49 8 of Spades
50 9 of Hearts
51 9 of Diamonds
52 9 of Clubs
53 9 of Spades
54 10 of Hearts
55 10 of Diamonds
56 10 of Clubs
57 10 of Spades
58 Jack of Hearts
59 Jack of Diamonds
60 Jack of Clubs
61 Jack of Spades
62 Queen of Hearts
63 Queen of Diamonds
64 Queen of Clubs
65 Queen of Spades
66 King of Hearts
67 King of Diamonds
68 King of Clubs
69 King of Spades
70
71 ***************************
72
73 1. Display the deck of cards
74 2. Shuffle the deck of cards
75 3. Draw a card from the deck
```

```
 4. Empty the deck
 5. Print a card from the deck
 6. Compare two cards
 7. Check if two cards are same
 8. Find card by rank and suit
 9. Deal a hand of cards
 10. Quit
 Enter your choice (1-10): 2
 **************************

 Deck shuffled.
 **************************

 1. Display the deck of cards
 2. Shuffle the deck of cards
 3. Draw a card from the deck
 4. Empty the deck
 5. Print a card from the deck
 6. Compare two cards
 7. Check if two cards are same
 8. Find card by rank and suit
 9. Deal a hand of cards
 10. Quit
 Enter your choice (1-10): 3
 **************************

 You drew: 2 of Clubs
 **************************

 1. Display the deck of cards
 2. Shuffle the deck of cards
 3. Draw a card from the deck
 4. Empty the deck
 5. Print a card from the deck
 6. Compare two cards
 7. Check if two cards are same
 8. Find card by rank and suit
 9. Deal a hand of cards
 10. Quit
 Enter your choice (1-10): 5
 **************************
```

```
117
118 Enter the position of the card you want to draw: 6
119 You drew: King of Diamonds
120 ***************************
121
122 1. Display the deck of cards
123 2. Shuffle the deck of cards
124 3. Draw a card from the deck
125 4. Empty the deck
126 5. Print a card from the deck
127 6. Compare two cards
128 7. Check if two cards are same
129 8. Find card by rank and suit
130 9. Deal a hand of cards
131 10. Quit
132 Enter your choice (1-10): 7
133 ***************************
134
135 You drew: 10 of Hearts and 9 of Diamonds and they
    are not ranked same.
136 ***************************
137
138 1. Display the deck of cards
139 2. Shuffle the deck of cards
140 3. Draw a card from the deck
141 4. Empty the deck
142 5. Print a card from the deck
143 6. Compare two cards
144 7. Check if two cards are same
145 8. Find card by rank and suit
146 9. Deal a hand of cards
147 10. Quit
148 Enter your choice (1-10): 8
149 ***************************
150
151 Enter the rank of the card you want to find: 9
152 Enter the suit ("Hearts", "Diamonds", "Clubs", "
    Spades") of the card you want to find: Hearts
153 Card found at position 36 in the deck.
154 ***************************
155
```

```
1. Display the deck of cards
2. Shuffle the deck of cards
3. Draw a card from the deck
4. Empty the deck
5. Print a card from the deck
6. Compare two cards
7. Check if two cards are same
8. Find card by rank and suit
9. Deal a hand of cards
10. Quit
Enter your choice (1-10): 4
**************************

Deck emptied.
**************************

1. Display the deck of cards
2. Shuffle the deck of cards
3. Draw a card from the deck
4. Empty the deck
5. Print a card from the deck
6. Compare two cards
7. Check if two cards are same
8. Find card by rank and suit
9. Deal a hand of cards
10. Quit
Enter your choice (1-10): 10
**************************

Goodbye!

Process finished with exit code 0

```