```
import java.util.*;
public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        Vector<Card> deck = Card.createDeck();
        int choice;
        do {
            System.out.println("1. Print the deck");
            System.out.println("2. Deal cards");
            System.out.println("3. Sort the deck");
            System.out.println("4. Search for a card");
            System.out.println("0. Exit");
            System.out.print("Enter your choice: ");
            choice = scanner.nextInt();
            switch (choice) {
                case 1:
                    Card.printDeck(deck);
                    break;
                case 2:
                    int numCards = scanner.nextInt();
                    Card.dealCards(deck, numCards);
                    break;
                case 3:
                    Card.sortDeck(deck);
                    System.out.println("Deck sorted!");
                    break;
                case 4:
                    System.out.print("Enter the rank of the card (0-12): ");
                    int rank = scanner.nextInt();
                    System.out.print("Enter the suit of the card (0-3): ");
                    int suit = scanner.nextInt();
                    Card card = new Card(rank, suit);
                    int index = Card.findCard(deck, card);
                    if (index == -1) {
                    System.out.println("Card not found in the deck!");
                    } else {
                    System.out.println("Card found at index " + index + " in the deck.");
                    }
                    break;
                case 0:
                    System.out.println("Exiting...");
                    break;
                default:
                    System.out.println("Invalid choice!");
             }
            System.out.println();
            } while (choice != 0);
        scanner.close();
    }
}
import java.util.*;
public class Card
{
    private int rank;
    private int suit;
    public Card() // Default constructor sets rank and suit to 0
    {
        this.rank = 0;
        this.suit = 0;
    }
    public Card(int rank, int suit)     // Parameterized constructor sets rank and suit to given values

    {
        this.rank = rank;
        this.suit = suit;
    }
    public int getRank()    // Getter methods for rank and suit
    {
        return rank;
    }
    public int getSuit()
    {
        return suit;
    }
    public void printCard()
    {
        String[] ranks = {"Ace", "2", "3", "4", "5", "6", "7", "8", "9", "10", "Jack", "Queen", "King"};
        String[] suits = {"Hearts", "Diamonds", "Clubs", "Spades"};
        System.out.println(ranks[rank] + " of " + suits[suit]);
    }
    public static Vector<Card> createDeck()
    {
        Vector<Card> deck = new Vector<Card>();
        for (int suit = 0; suit < 4; suit++)
        {
            for (int rank = 0; rank < 13; rank++)
            {
                deck.add(new Card(rank, suit));
            }
        }
        return deck;
    }
    public static void printDeck(Vector<Card> deck)
    {
        for (int i = 0; i < deck.size(); i++)
        {
            deck.get(i).printCard();
        }
    }
    public static boolean sameCard(Card c1, Card c2)
    {
        return c1.getRank() == c2.getRank() && c1.getSuit() == c2.getSuit();
    }
    public static int compareCard(Card c1, Card c2)
    {
        if (c1.getRank() < c2.getRank())
        {
            return -1;
        }
        else if (c1.getRank() > c2.getRank())
        {
            return 1;
        }
        else
        {
            return Integer.compare(c1.getSuit(), c2.getSuit());
        }
    }
    public static void sortDeck(Vector<Card> deck)
    {
        Collections.sort(deck, new Comparator<Card>()
        {
            @Override
            public int compare(Card c1, Card c2)
            {
                return compareCard(c1, c2);
            }
        });
    }

    public static int findCard(Vector<Card> deck, Card card)
    {
        for (int i = 0; i < deck.size(); i++)
        {
            if (sameCard(deck.get(i), card))
            {
                return i;
            }
        }
        return -1;
    }
    public static void dealCards(Vector<Card> deck, int numCards)
    {
        Random rand = new Random();
        for (int i = 0; i < numCards; i++) {
            int index = rand.nextInt(deck.size());
            Card card = deck.remove(index);
            card.printCard();
        }
    }
}
```

https://github.com/adityapande403/java4

```
"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
**************************

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): 1
**************************

Deck of Cards:
Ace of Hearts
Ace of Diamonds
Ace of Clubs
Ace of Spades
2 of Hearts
2 of Diamonds
2 of Clubs
2 of Spades
3 of Hearts
3 of Diamonds
3 of Clubs
3 of Spades
4 of Hearts
4 of Diamonds
4 of Clubs
4 of Spades
5 of Hearts
```

```
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

```

```
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
```

```
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
**************************
```

```
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
```