

# JAVA ASSIGNMENT 4

NAME : ANANYA KUMAR

PRN : 21070126012

BATCH : AIML A1

## 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)

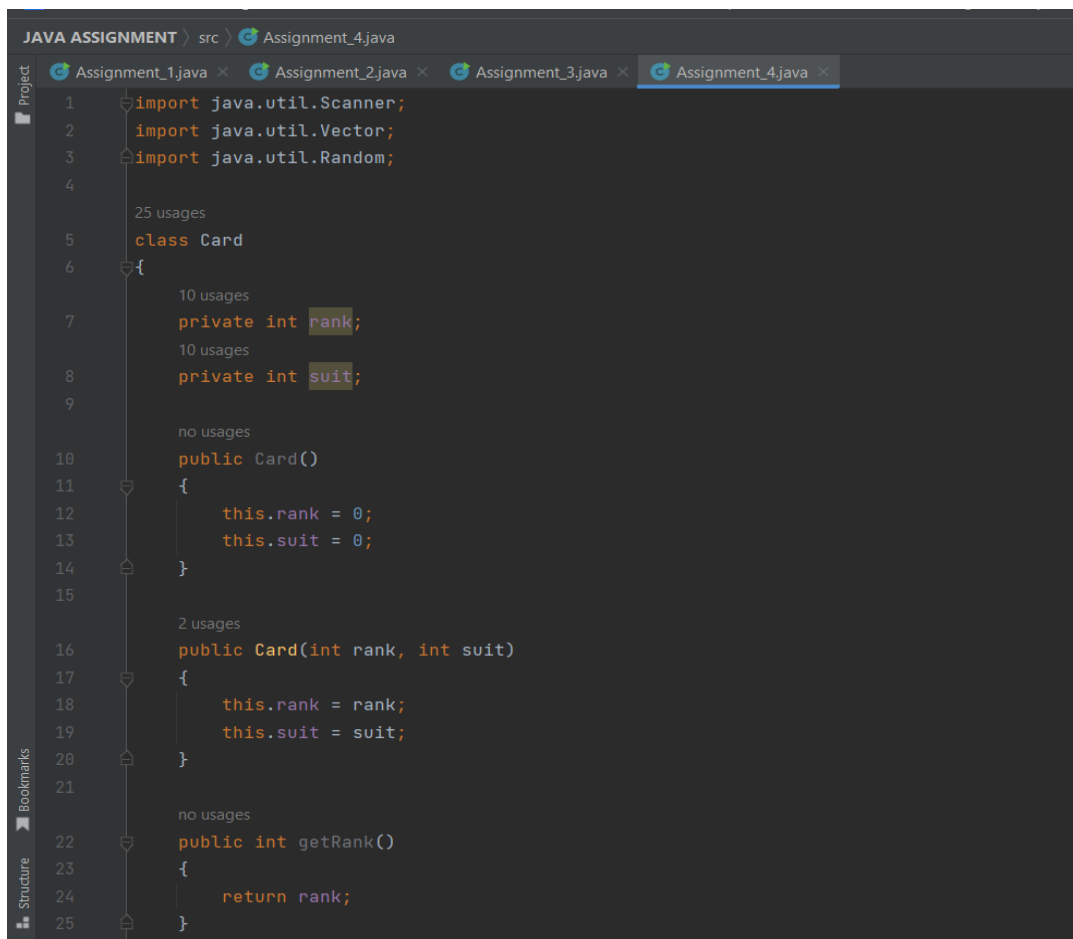
1. Use integers to encode the ranks and suits.
2. Have suitable default & parameterized constructors.
3. all data members to have private access.
4. The class 'Card' to have the following methods:

createDeck(), printCard(), printDeck (),sameCard(),compareCard(), sortCard(), findCard()

which searches through an array or vector of Cards to see whether it contains a certain card,

dealCards() function: to print 5 random cards from the existing deck.

## CODE :



```
1 import java.util.Scanner;
2 import java.util.Vector;
3 import java.util.Random;
4
5 class Card
6 {
7     private int rank;
8     private int suit;
9
10    public Card()
11    {
12        this.rank = 0;
13        this.suit = 0;
14    }
15
16    public Card(int rank, int suit)
17    {
18        this.rank = rank;
19        this.suit = suit;
20    }
21
22    public int getRank()
23    {
24        return rank;
25    }
```

```
JAVA ASSIGNMENT > src > Assignment_4.java > Card > compareCard
Assignment_1.java x Assignment_2.java x Assignment_3.java x Assignment_4.java x
54         for (Card card : deck)
55         {
56             card.printCard();
57         }
58     }
59
60     2 usages
61     @ public boolean sameCard(Card other)
62     {
63         return (this.rank == other.rank && this.suit == other.suit);
64     }
65
66     1 usage
67     @ public int compareCard(Card other)
68     {
69         if (this.rank < other.rank)
70         {
71             return -1;
72         } else if (this.rank > other.rank)
73         {
74             return 1;
75         } else
76         {
77             if (this.suit < other.suit)
78             {
79                 return -1;
80             } else if (this.suit > other.suit)
81             {
82                 return 1;
83             } else
84             {
85                 return 0;
86             }
87         }
88     }
```

```
JAVA ASSIGNMENT > src > Assignment_4.java
Assignment_1.java x Assignment_2.java x Assignment_3.java x Assignment_4.java x
27     public int getSuit()
28     {
29         return suit;
30     }
31
32     2 usages
33     public void printCard()
34     {
35         String[] suits = {"Spades", "Hearts", "Diamonds", "Clubs"};
36         String[] ranks = {"Ace", "2", "3", "4", "5", "6", "7", "8", "9", "10", "Jack", "Queen", "King"};
37         System.out.println(ranks[this.rank] + " of " + suits[this.suit]);
38     }
39
40     1 usage
41     @ public static Vector<Card> createDeck()
42     {
43         Vector<Card> deck = new Vector<Card>();
44         for (int suit = 0; suit < 4; suit++)
45         {
46             for (int rank = 0; rank < 13; rank++)
47             {
48                 deck.add(new Card(rank, suit));
49             }
50         }
51         return deck;
52     }
53
54     2 usages
55     @ public static void printDeck(Vector<Card> deck)
56     {
57         for (Card card : deck)
58         {
59             card.printCard();
60         }
61     }
```

```
JAVA ASSIGNMENT > src > Assignment_4.java > Card > compareCard
Assignment_1.java x Assignment_2.java x Assignment_3.java x Assignment_4.java x
82 {
83     return 0;
84 }
85 }
86 }
87
1 usage
88 @ public static void sortDeck(Vector<Card> deck)
89 {
90     deck.sort((c1, c2) -> c1.compareCard(c2));
91 }
92
1 usage
93 @ public static void findCard(Vector<Card> deck, Card card)
94 {
95     for (int i = 0; i < deck.size(); i++)
96     {
97         if (deck.get(i).sameCard(card))
98         {
99             System.out.println("Card found at index " + i);
100             return;
101         }
102     }
103     System.out.println("Card not found");
104 }
105
1 usage
106 public static void dealCards(Vector<Card> deck, int numCards)
107 {
108     Random rand = new Random();
109     for (int i = 0; i < numCards; i++)
```

```
JAVA ASSIGNMENT > src > Assignment_4.java > Card > compareCard
Assignment_1.java x Assignment_2.java x Assignment_3.java x Assignment_4.java x
109     for (int i = 0; i < numCards; i++)
110     {
111         int index = rand.nextInt(deck.size());
112         Card card = deck.get(index);
113         card.printCard();
114         deck.remove(index);
115     }
116 }
117 }
118
no usages
119 public class Assignment_4
120 {
121     no usages
122     public static void main(String[] args)
123     {
124         Scanner input = new Scanner(System.in);
125         Vector<Card> deck = Card.createDeck();
126
127         while (true)
128         {
129             System.out.println("\n--- Menu ---");
130             System.out.println("1. Print the deck");
131             System.out.println("2. Sort the deck");
132             System.out.println("3. Check if two cards are the same");
133             System.out.println("4. Find a card");
134             System.out.println("5. Deal cards");
135             System.out.println("6. Exit");
136             System.out.print("Enter your choice (1-6): ");
137             int choice = input.nextInt();
```

```
JAVA ASSIGNMENT > src > Assignment_4.java > Card > compareCard
Assignment_1.java x Assignment_2.java x Assignment_3.java x Assignment_4.java x
137
138     if (choice == 1)
139     {
140         System.out.println("\n--- Deck ---");
141         Card.printDeck(deck);
142     }
143
144     else if (choice == 2)
145     {
146         Card.sortDeck(deck);
147         System.out.println("\n--- Sorted deck ---");
148         Card.printDeck(deck);
149     }
150
151     else if (choice == 3)
152     {
153         System.out.println("\nEnter the first card:");
154         Card card1 = readCard(input);
155         System.out.println("Enter the second card:");
156         Card card2 = readCard(input);
157         if (card1.sameCard(card2))
158         {
159             System.out.println("The two cards are the same");
160         }
161         else
162         {
163             System.out.println("The two cards are different");
164         }
165     }
166
167     else if (choice == 4)
```

```
JAVA ASSIGNMENT > src > Assignment_4.java > Card > compareCard
Assignment_1.java x Assignment_2.java x Assignment_3.java x Assignment_4.java x
167
168     else if (choice == 4)
169     {
170         System.out.println("\nEnter a card to search for:");
171         Card card = readCard(input);
172         Card.findCard(deck, card);
173     }
174
175     else if (choice == 5)
176     {
177         System.out.println("\nDealing cards...");
178         Card.dealCards(deck, numCards: 5);
179     }
180
181     else if (choice == 6)
182     {
183         System.out.println("Thank You!");
184         break;
185     }
186
187     else
188     {
189         System.out.println("Invalid choice, please try again");
190     }
191 }
192
3 usages
193 @
194 public static Card readCard(Scanner input)
195 {
196     System.out.print("Enter rank (0-12): ");
197     int rank = input.nextInt();
```

```

3 usages
193 @      public static Card readCard(Scanner input)
194      {
195          System.out.print("Enter rank (0-12): ");
196          int rank = input.nextInt();
197          System.out.print("Enter suit (0-3): ");
198          int suit = input.nextInt();
199          return new Card(rank, suit);
200      }
201  }
202

```

## OUTPUT :

```

JAVA ASSIGNMENT > src > Assignment_4.java > Card > compareCard
Assignment_1.java x Assignment_2.java x Assignment_3.java x Assignment_4.java x
Run: Assignment_4 x
"C:\Program Files\Java\jdk-19\bin\java.exe" "-javaagent:C:\Program
--- Menu ---
1. Print the deck
2. Sort the deck
3. Check if two cards are the same
4. Find a card
5. Deal cards
6. Exit
Enter your choice (1-6): 1

--- Deck ---
Ace of Spades
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 Hearts
2 of Hearts
3 of Hearts
4 of Hearts
5 of Hearts

```

JAVA ASSIGNMENT > src > Assignment\_4.java > Card > compareCard

Assignment\_1.java x Assignment\_2.java x Assignment\_3.java x Assignment\_4.java x

Run: Assignment\_4 x

```
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 Diamonds
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 Clubs
2 of Clubs
3 of Clubs
4 of Clubs
5 of Clubs
6 of Clubs
7 of Clubs
```

JAVA ASSIGNMENT > src > Assignment\_4.java > Card > compareCard

Assignment\_1.java x Assignment\_2.java x Assignment\_3.java x Assignment\_4.java x

Run: Assignment\_4 x

```
7 of Clubs
8 of Clubs
9 of Clubs
10 of Clubs
Jack of Clubs
Queen of Clubs
King of Clubs
```

```
JAVA ASSIGNMENT  src  Assignment_4.java  Card  compareCard
Assignment_1.java  Assignment_2.java  Assignment_3.java  Assignment_4.java
Run:  Assignment_4
5 of Clubs
6 of Spades
6 of Hearts
6 of Diamonds
6 of Clubs
7 of Spades
7 of Hearts
7 of Diamonds
7 of Clubs
8 of Spades
8 of Hearts
8 of Diamonds
8 of Clubs
9 of Spades
9 of Hearts
9 of Diamonds
9 of Clubs
10 of Spades
10 of Hearts
10 of Diamonds
10 of Clubs
Jack of Spades
Jack of Hearts
Jack of Diamonds
Jack of Clubs
Queen of Spades
Queen of Hearts
Queen of Diamonds
Queen of Clubs
King of Spades
```

```
JAVA ASSIGNMENT  src  Assignment_4.java  Card  compareCard
Assignment_1.java  Assignment_2.java  Assignment_3.java  Assignment_4.java
Run:  Assignment_4
--- Menu ---
1. Print the deck
2. Sort the deck
3. Check if two cards are the same
4. Find a card
5. Deal cards
6. Exit
Enter your choice (1-6): 2

--- Sorted deck ---
Ace of Spades
Ace of Hearts
Ace of Diamonds
Ace of Clubs
2 of Spades
2 of Hearts
2 of Diamonds
2 of Clubs
3 of Spades
3 of Hearts
3 of Diamonds
3 of Clubs
4 of Spades
4 of Hearts
4 of Diamonds
4 of Clubs
5 of Spades
5 of Hearts
5 of Diamonds
```

```
JAVA ASSIGNMENT > src > Assignment_4.java > Card > compareCard
Assignment_1.java x Assignment_2.java x Assignment_3.java x Assignment_4.java x
Run: Assignment_4 x
7 of Hearts
7 of Diamonds
7 of Clubs
8 of Spades
8 of Hearts
8 of Diamonds
8 of Clubs
9 of Spades
9 of Hearts
9 of Diamonds
9 of Clubs
10 of Spades
10 of Hearts
10 of Diamonds
10 of Clubs
Jack of Spades
Jack of Hearts
Jack of Diamonds
Jack of Clubs
Queen of Spades
Queen of Hearts
Queen of Diamonds
Queen of Clubs
King of Spades
King of Hearts
King of Diamonds
King of Clubs
```

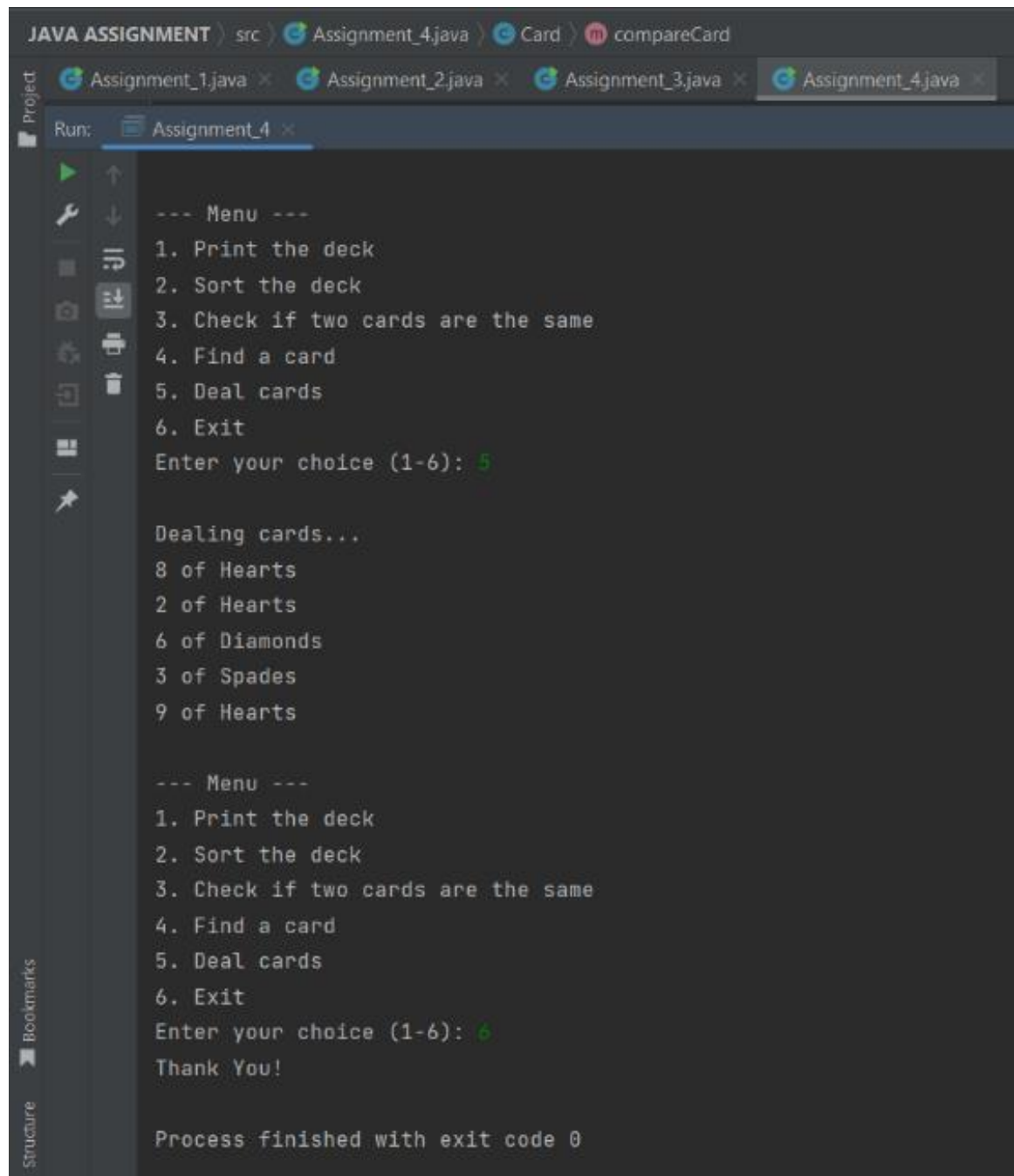
```
JAVA ASSIGNMENT > src > Assignment_4.java > Card > compareCard
Assignment_1.java x Assignment_2.java x Assignment_3.java x Assignment_4.java x
Run: Assignment_4 x
--- Menu ---
1. Print the deck
2. Sort the deck
3. Check if two cards are the same
4. Find a card
5. Deal cards
6. Exit
Enter your choice (1-6): 3

Enter the first card:
Enter rank (0-12): 4
Enter suit (0-3): 2
Enter the second card:
Enter rank (0-12): 12
Enter suit (0-3): 1
The two cards are different

--- Menu ---
1. Print the deck
2. Sort the deck
3. Check if two cards are the same
4. Find a card
5. Deal cards
6. Exit
Enter your choice (1-6): 4

Enter a card to search for:
Enter rank (0-12): 8
Enter suit (0-3): 1
Card found at index 33
```





```
JAVA ASSIGNMENT > src > Assignment_4.java > Card > compareCard
Assignment_1.java x Assignment_2.java x Assignment_3.java x Assignment_4.java x
Run: Assignment_4 x
--- Menu ---
1. Print the deck
2. Sort the deck
3. Check if two cards are the same
4. Find a card
5. Deal cards
6. Exit
Enter your choice (1-6): 5

Dealing cards...
8 of Hearts
2 of Hearts
6 of Diamonds
3 of Spades
9 of Hearts

--- Menu ---
1. Print the deck
2. Sort the deck
3. Check if two cards are the same
4. Find a card
5. Deal cards
6. Exit
Enter your choice (1-6): 6

Thank You!

Process finished with exit code 0
```

## GITHUB LINK :

[https://github.com/ananya2285/Java-Assignment-/blob/c903547e65d4042dea84001c1e63c413a9d577fa/Assignment\\_4.java](https://github.com/ananya2285/Java-Assignment-/blob/c903547e65d4042dea84001c1e63c413a9d577fa/Assignment_4.java)