JAVA ASSIGNMENT 4

NAME : ANVESHA RAIKWAR

PRN : 21070126017

BATCH : AIML A1

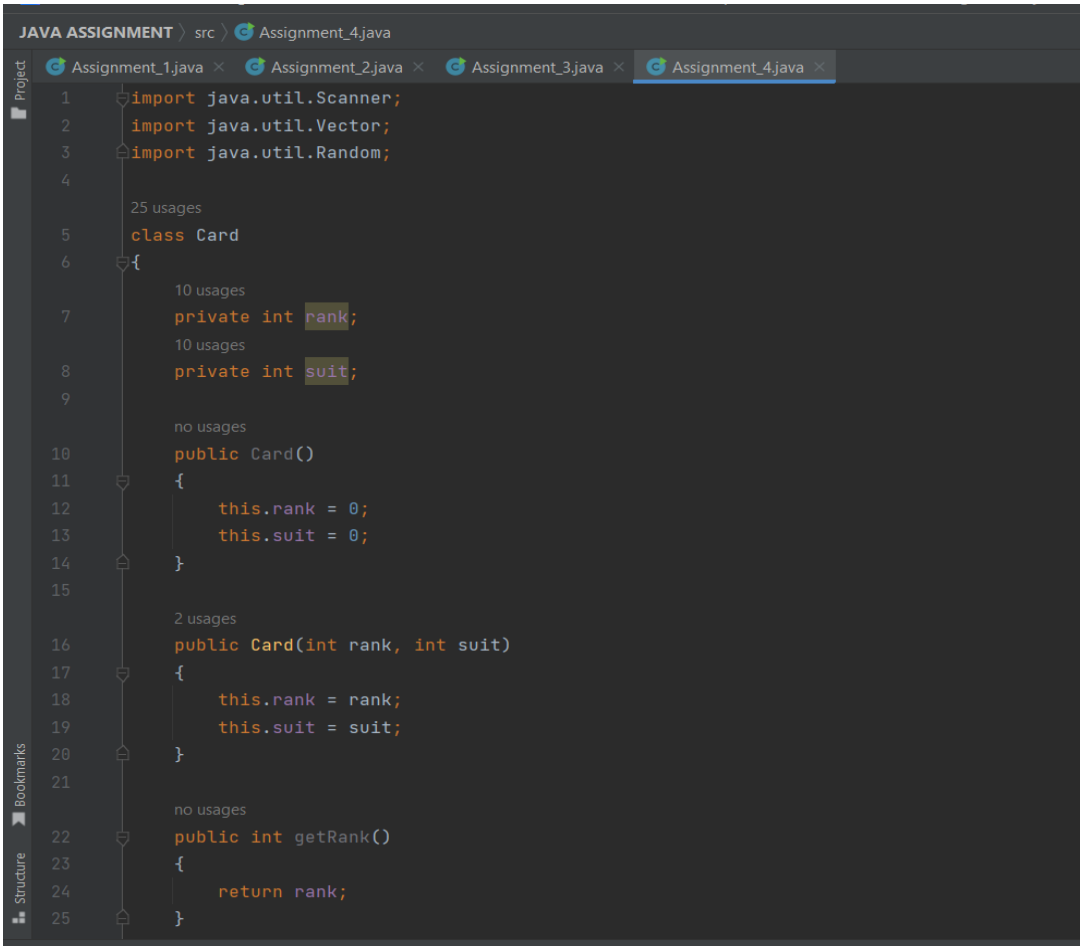
QUESTION :

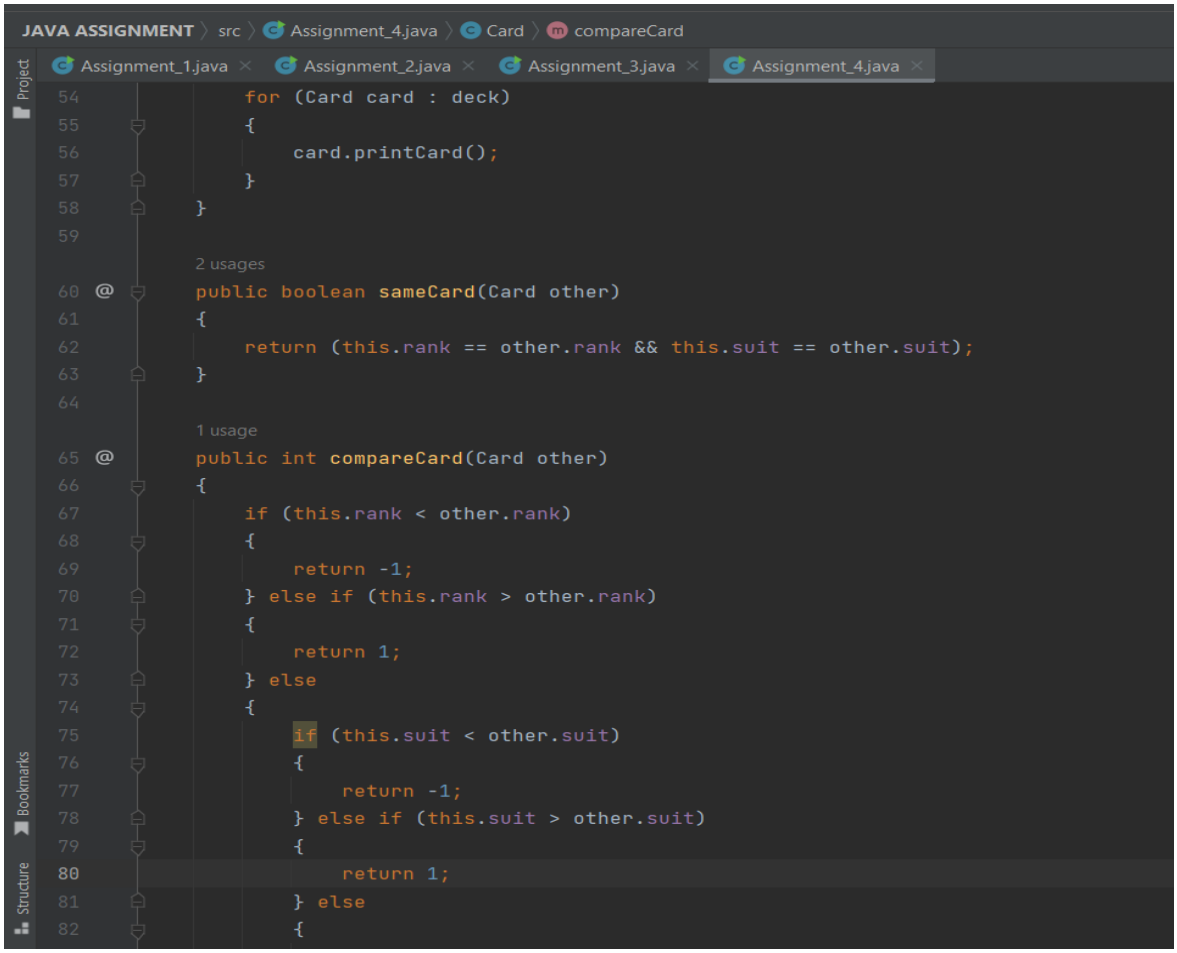
Write a menu-driven Java Program for the following:
There are 52 cards in a deck, each of which belongs to one of four suits and one of 13 ranks. Represent a deck of cards as an array of
Objects (*you may use the Vector class)

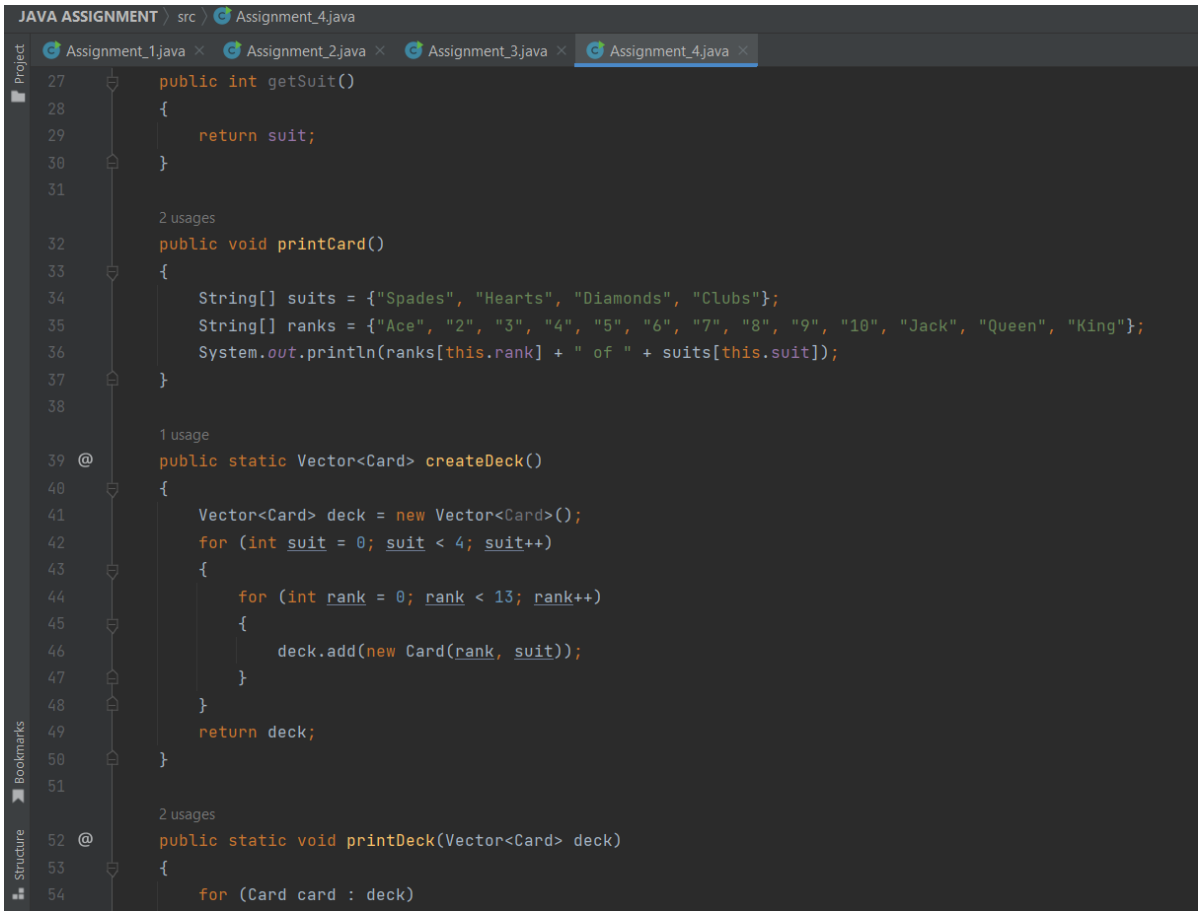
1. Use integers to encode the ranks and suits.
2. Have suitable default & parameterized constructors.
3. all data members to have private access.
4. The class 'Card' to have the following methods:

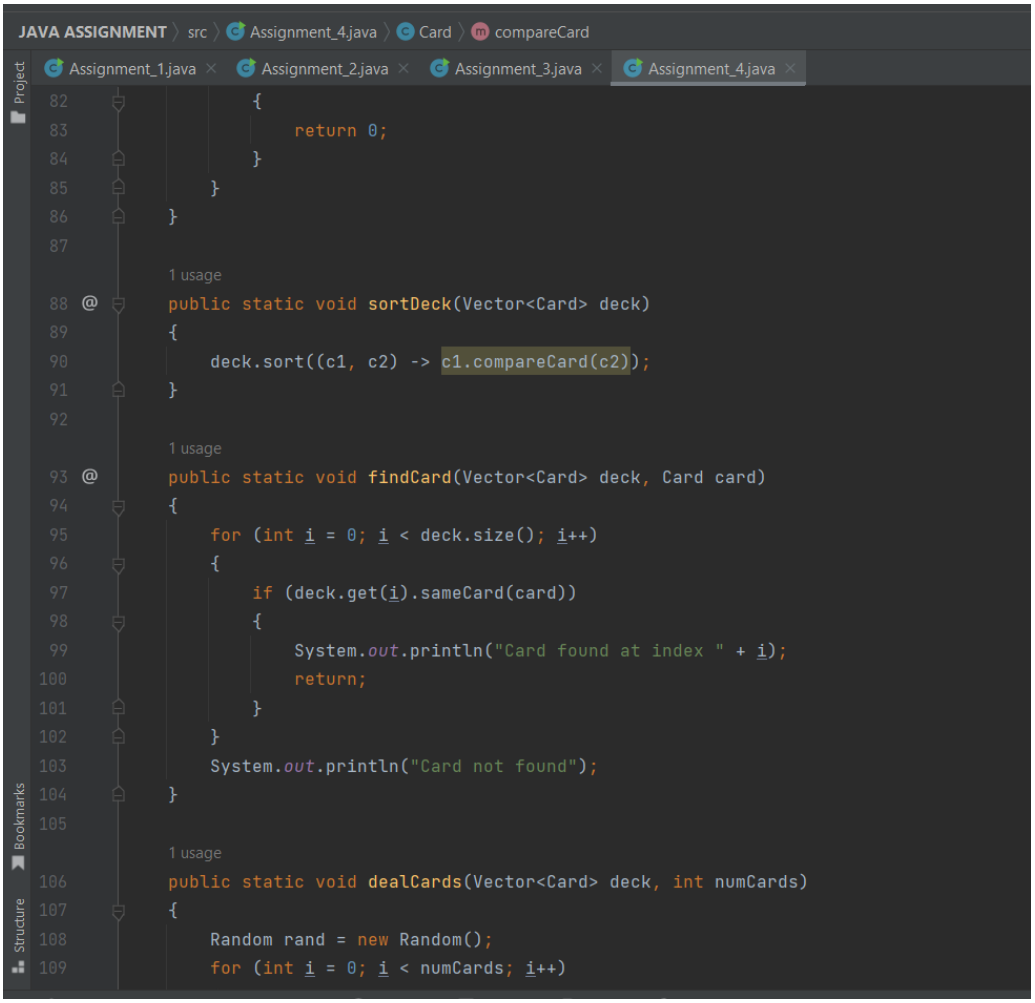
createDeck(), printCard(), printDeck (),sameCard(),compareCard(), sortCard(), findCard() which searches through an array or vector of Cards to see whether it contains a certain card, dealCards() function: to print 5 random cards from the existing deck.

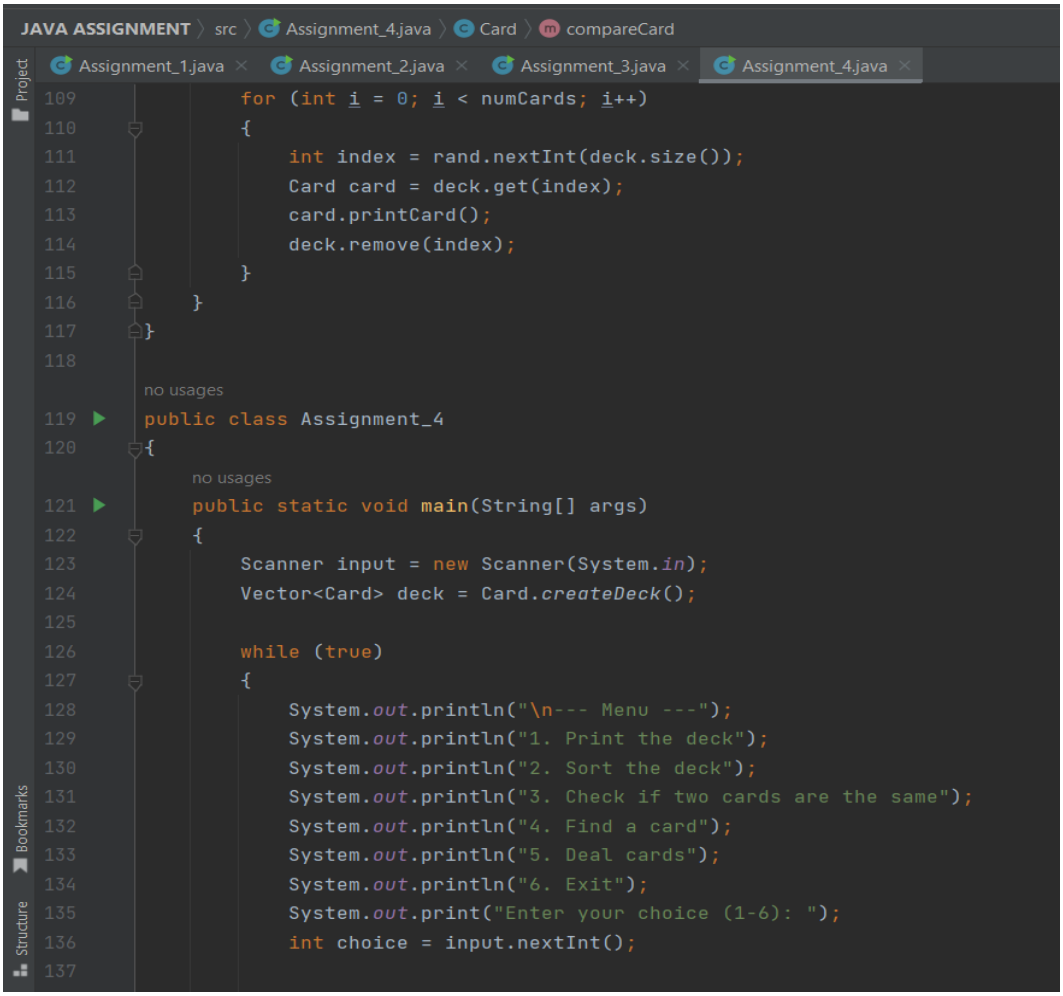
CODE :

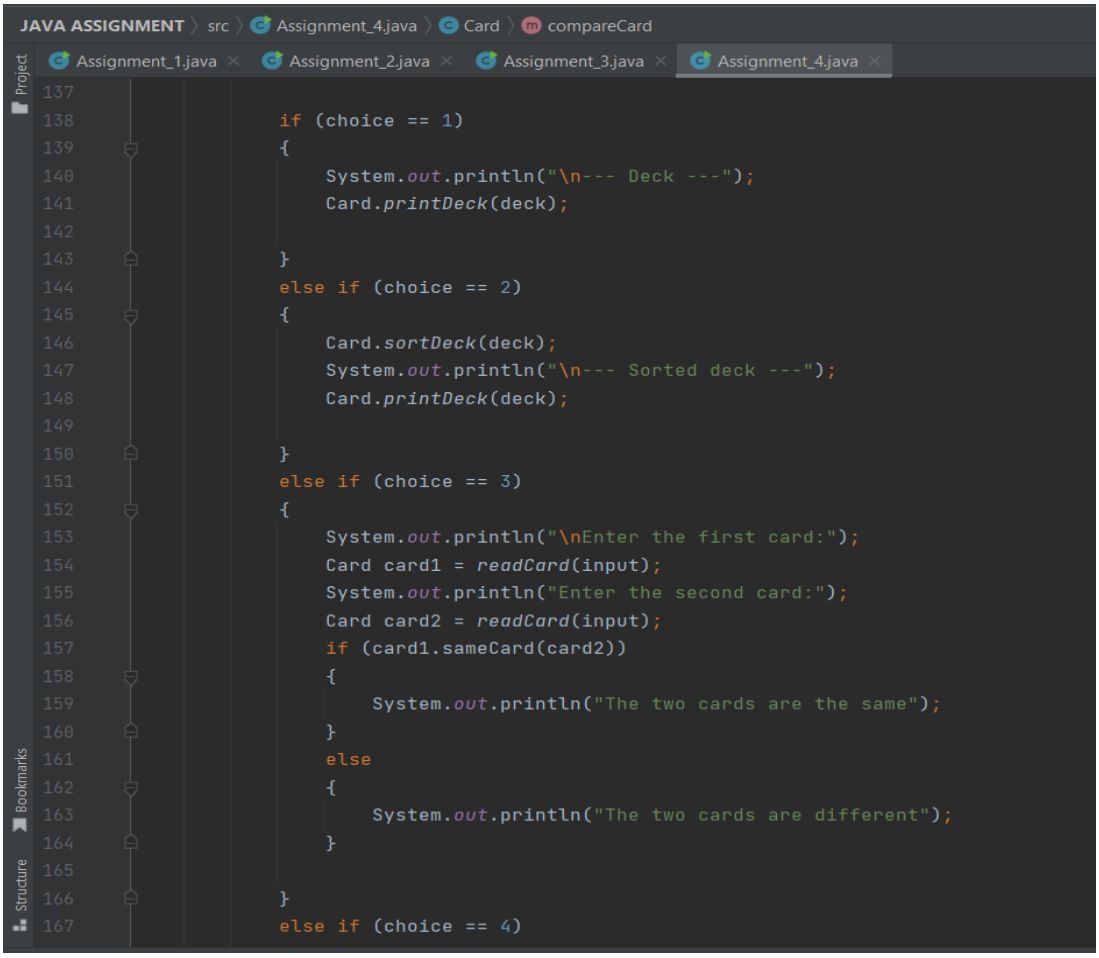


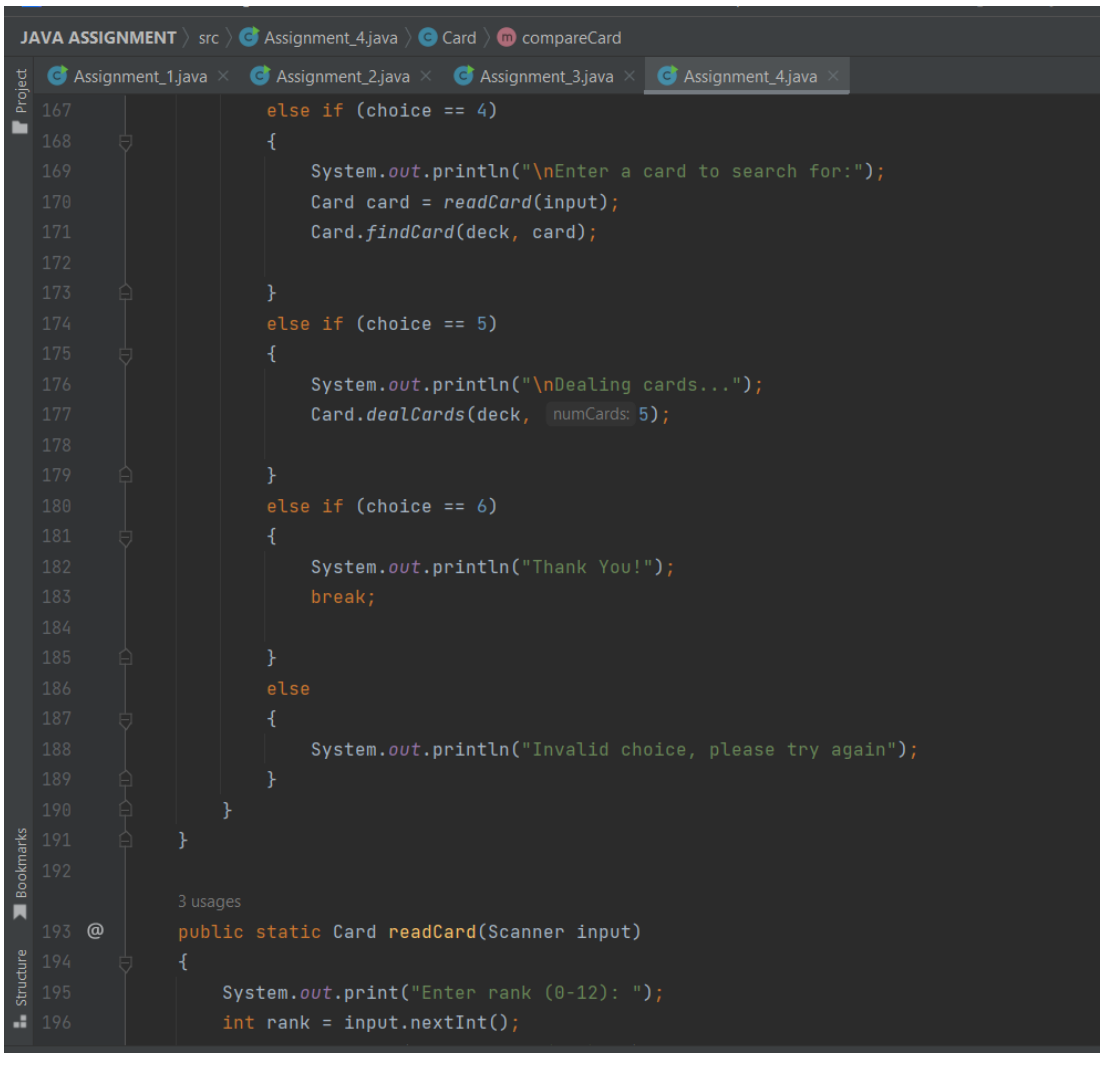

JAVA ASSIGNMENT src Assignment_4.java Card compareCard
Assignment_1.java Assignment_2.java Assignment_3.java Assignment_4.java
Project
Bookmarks
Structure
for (Card card : deck)
{
card.printCard();
}
}
2 usages
public boolean sameCard(Card other)
{
return (this.rank == other.rank && this.suit == other.suit);
}
1 usage
public int compareCard(Card other)
{
if (this.rank < other.rank)
{
return -1;
} else if (this.rank > other.rank)
{
return 1;
} else
{
if (this.suit < other.suit)
{
return -1;
} else if (this.suit > other.suit)
{
return 1;
} else
{


JAVA ASSIGNMENT src Assignment_4.java
Assignment_1.java Assignment_2.java Assignment_3.java Assignment_4.java
Project
Bookmarks
Structure
public int getSuit()
{
return suit;
}
2 usages
public void printCard()
{
String[] suits = {"Spades", "Hearts", "Diamonds", "Clubs"};
String[] ranks = {"Ace", "2", "3", "4", "5", "6", "7", "8", "9", "10", "Jack", "Queen", "King"};
System.out.println(ranks[this.rank] + " of " + suits[this.suit]);
}
1 usage
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;
}
2 usages
public static void printDeck(Vector<Card> deck)
{
for (Card card : deck)


JAVA ASSIGNMENT src Assignment_4.java Card compareCard
Assignment_1.java Assignment_2.java Assignment_3.java Assignment_4.java
Project
Bookmarks
Structure
{
return 0;
}
}
}
1 usage
public static void sortDeck(Vector<Card> deck)
{
deck.sort((c1, c2) -> c1.compareCard(c2));
}
1 usage
public static void findCard(Vector<Card> deck, Card card)
{
for (int i = 0; i < deck.size(); i++)
{
if (deck.get(i).sameCard(card))
{
System.out.println("Card found at index " + i);
return;
}
}
System.out.println("Card not found");
}
1 usage
public static void dealCards(Vector<Card> deck, int numCards)
{
Random rand = new Random();
for (int i = 0; i < numCards; i++)

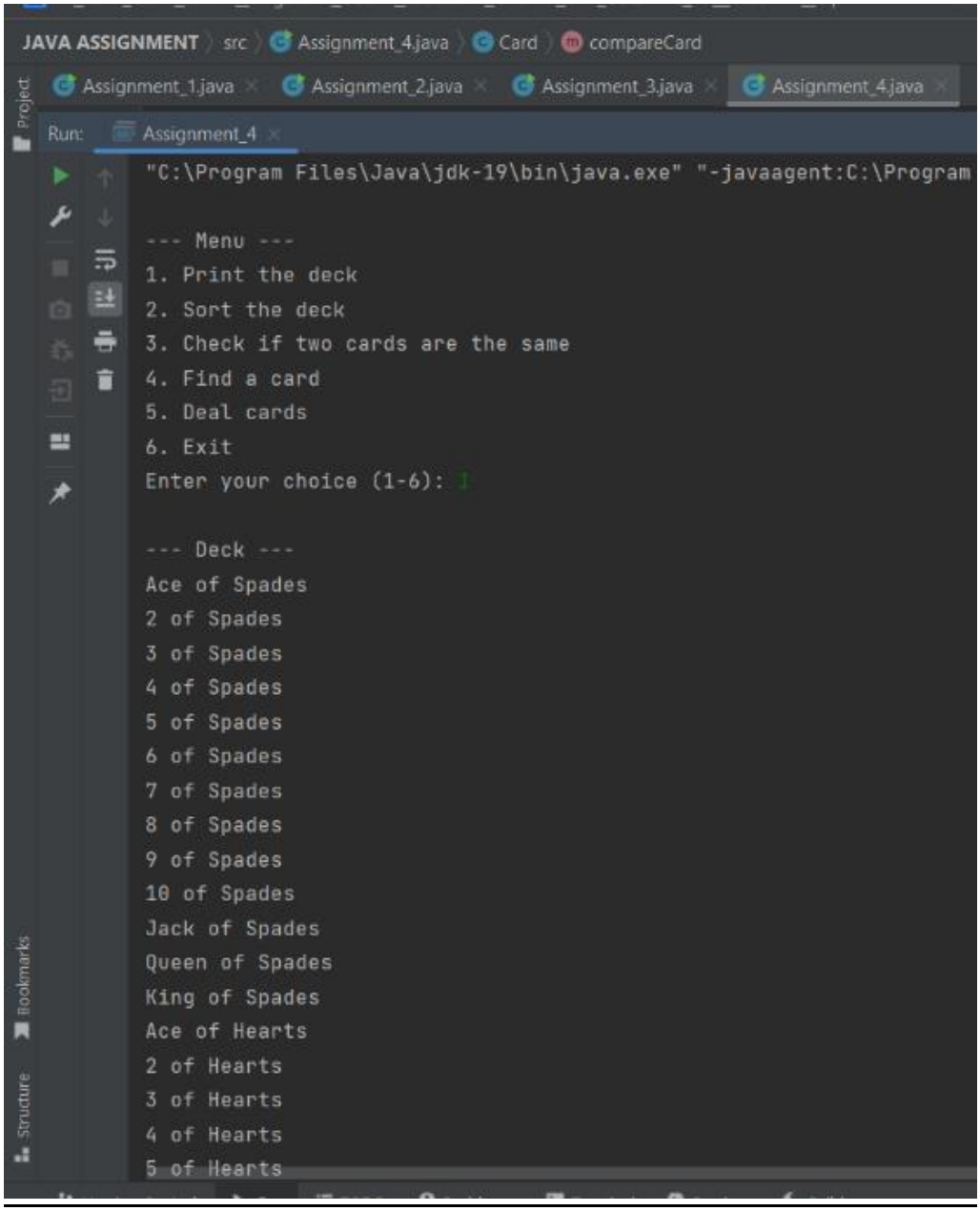

JAVA ASSIGNMENT src Assignment_4.java Card compareCard
Assignment_1.java Assignment_2.java Assignment_3.java Assignment_4.java
Project
Bookmarks
Structure
for (int i = 0; i < numCards; i++)
{
int index = rand.nextInt(deck.size());
Card card = deck.get(index);
card.printCard();
deck.remove(index);
}
}
}
no usages
public class Assignment_4
{
no usages
public static void main(String[] args)
{
Scanner input = new Scanner(System.in);
Vector<Card> deck = Card.createDeck();
while (true)
{
System.out.println("\n--- Menu ---");
System.out.println("1. Print the deck");
System.out.println("2. Sort the deck");
System.out.println("3. Check if two cards are the same");
System.out.println("4. Find a card");
System.out.println("5. Deal cards");
System.out.println("6. Exit");
System.out.print("Enter your choice (1-6): ");
int choice = input.nextInt();

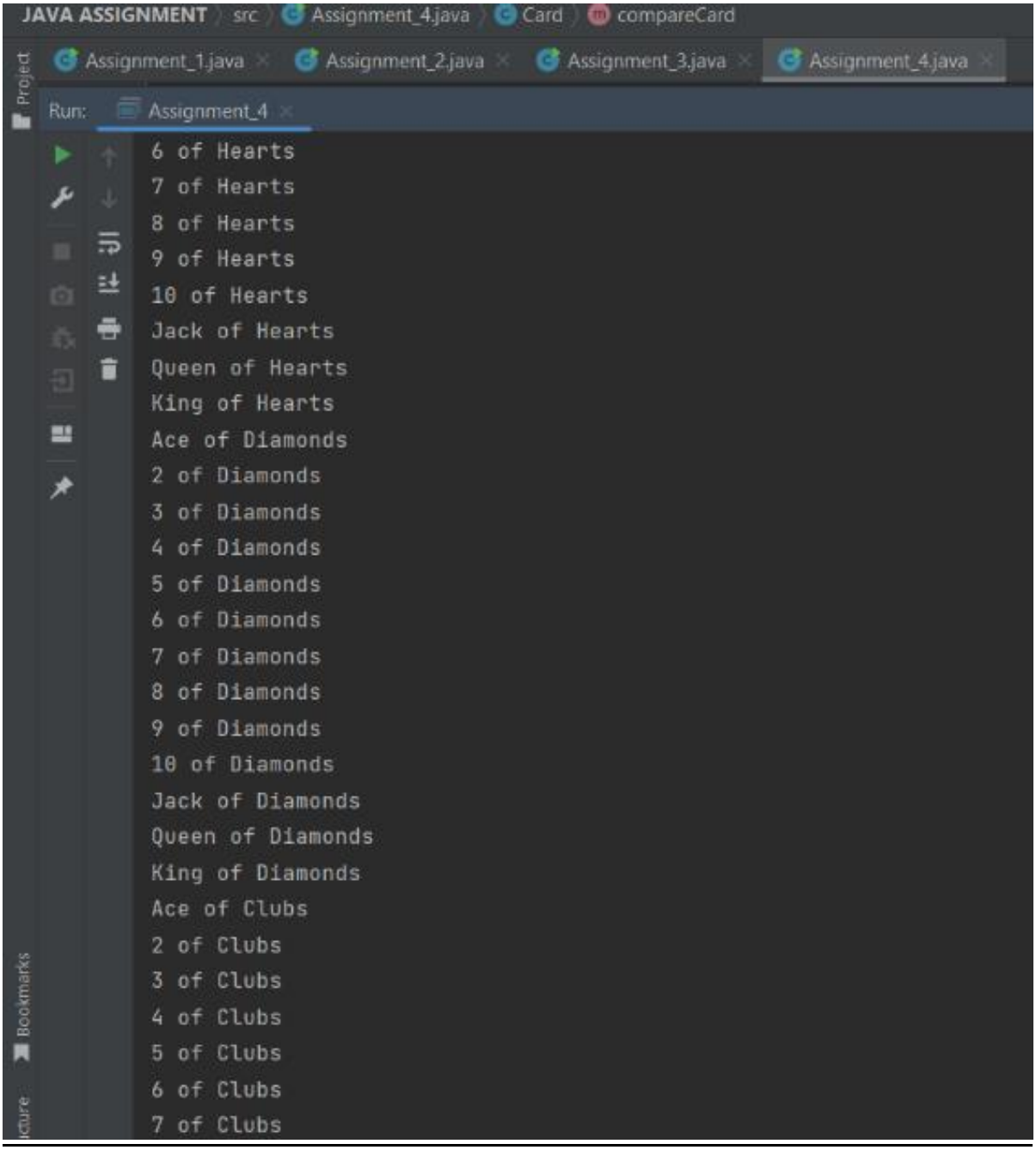

JAVA ASSIGNMENT src Assignment_4.java Card compareCard
Assignment_1.java Assignment_2.java Assignment_3.java Assignment_4.java
Project
Bookmarks
Structure
137
138 if (choice == 1)
139 {
140 System.out.println("\n--- Deck ---");
141 Card.printDeck(deck);
142
143 }
144 else if (choice == 2)
145 {
146 Card.sortDeck(deck);
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)

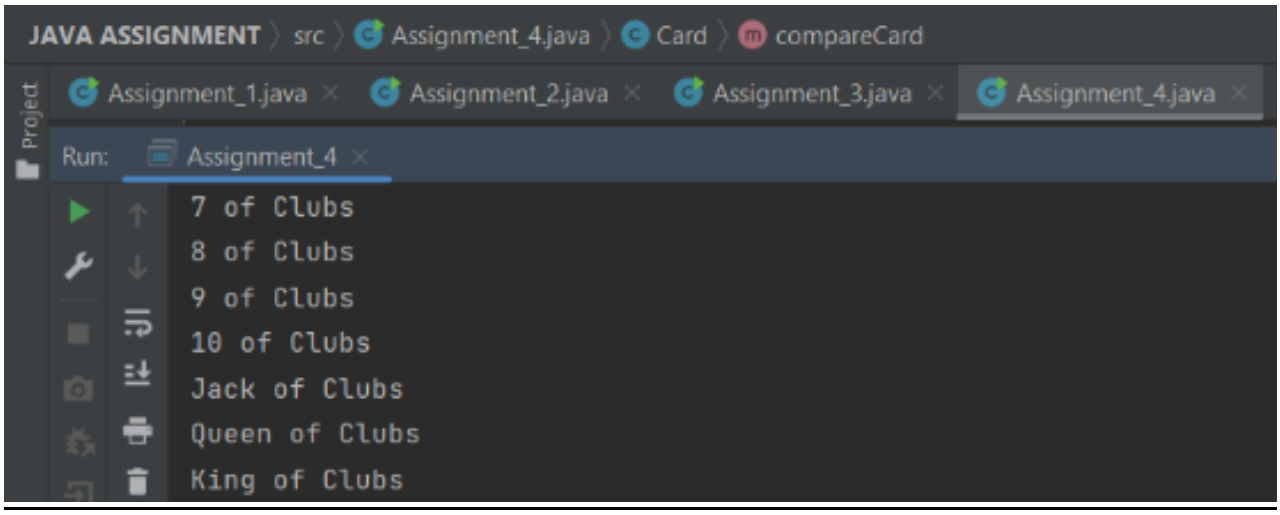

JAVA ASSIGNMENT src Assignment_4.java Card compareCard
Assignment_1.java Assignment_2.java Assignment_3.java Assignment_4.java
Project
Bookmarks
Structure
167 else if (choice == 4)
168 {
169 System.out.println("\nEnter a card to search for:");
170 Card card = readCard(input);
171 Card.findCard(deck,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

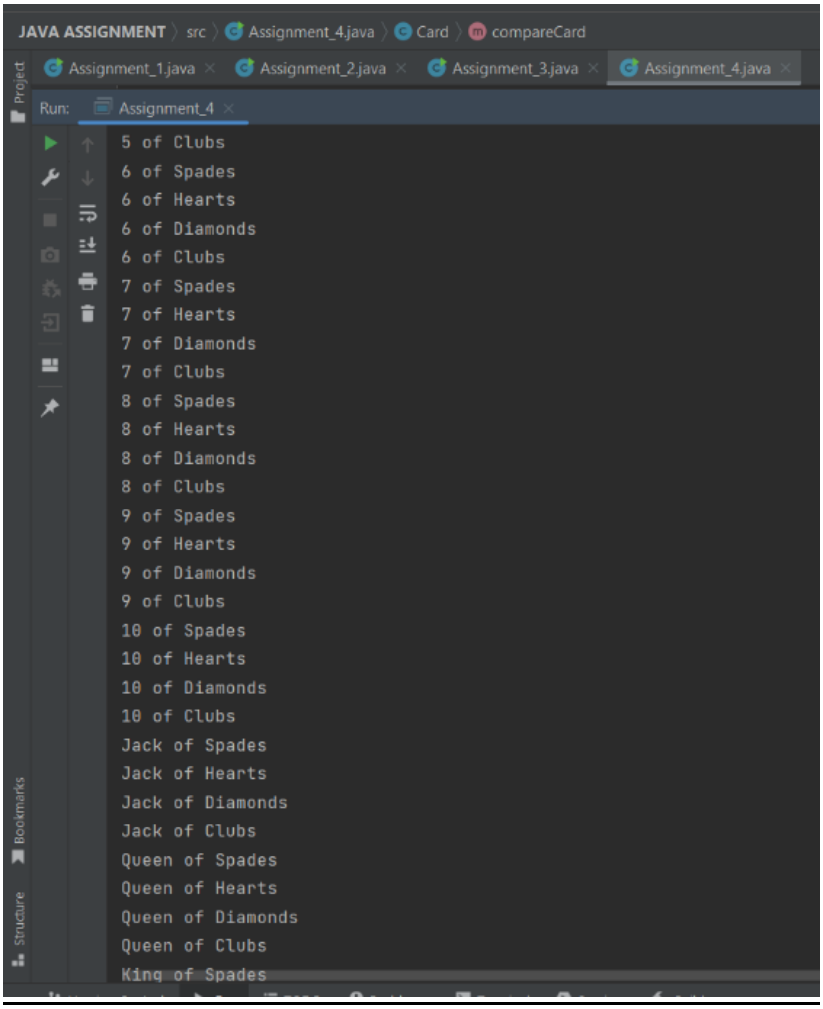
```
        3 usages
193 @   public static Card readCard(Scanner input)
194     {
195         System.out.print("Enter rank (0-12): ");
196         int rank = input.nextInt();
197         System.out.print("Enter suit (0-3): ");
198         int suit = input.nextInt();
199         return new Card(rank,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

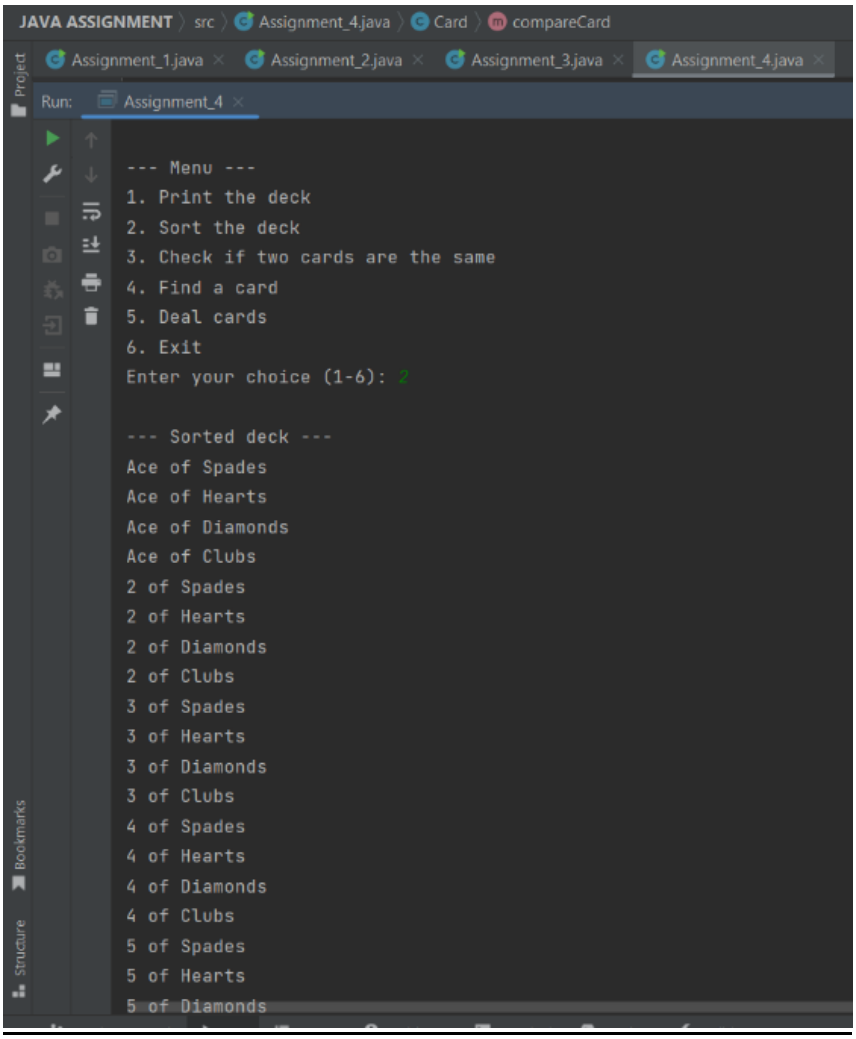
OUTPUT :

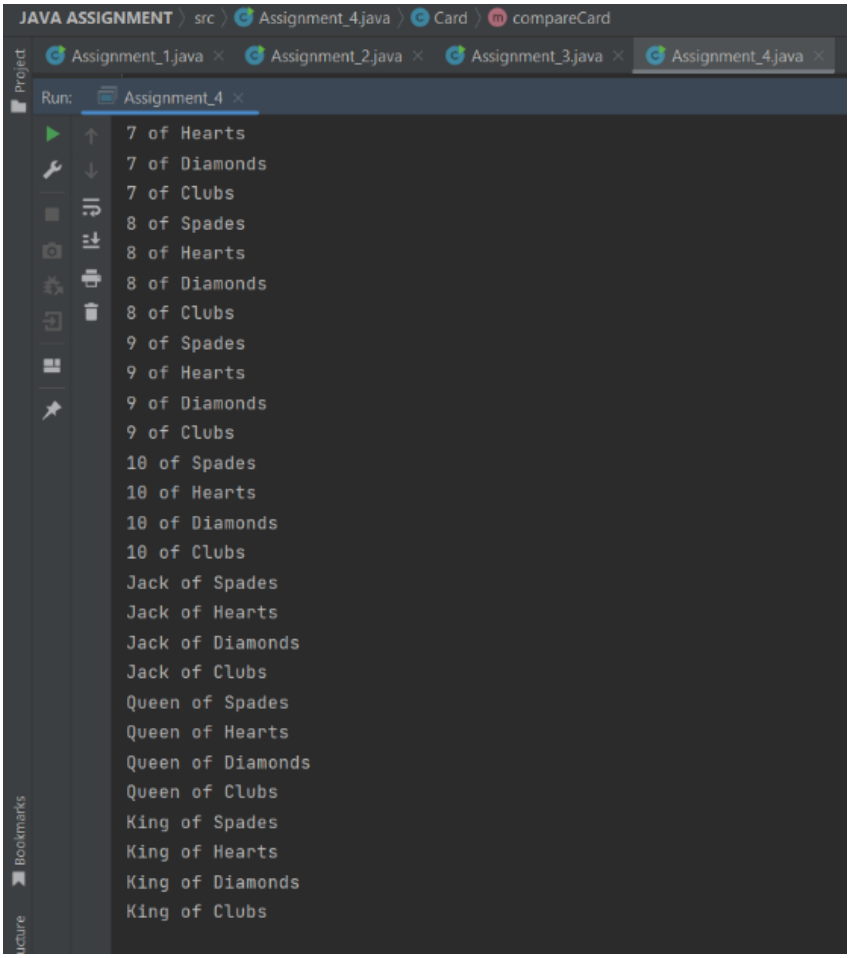


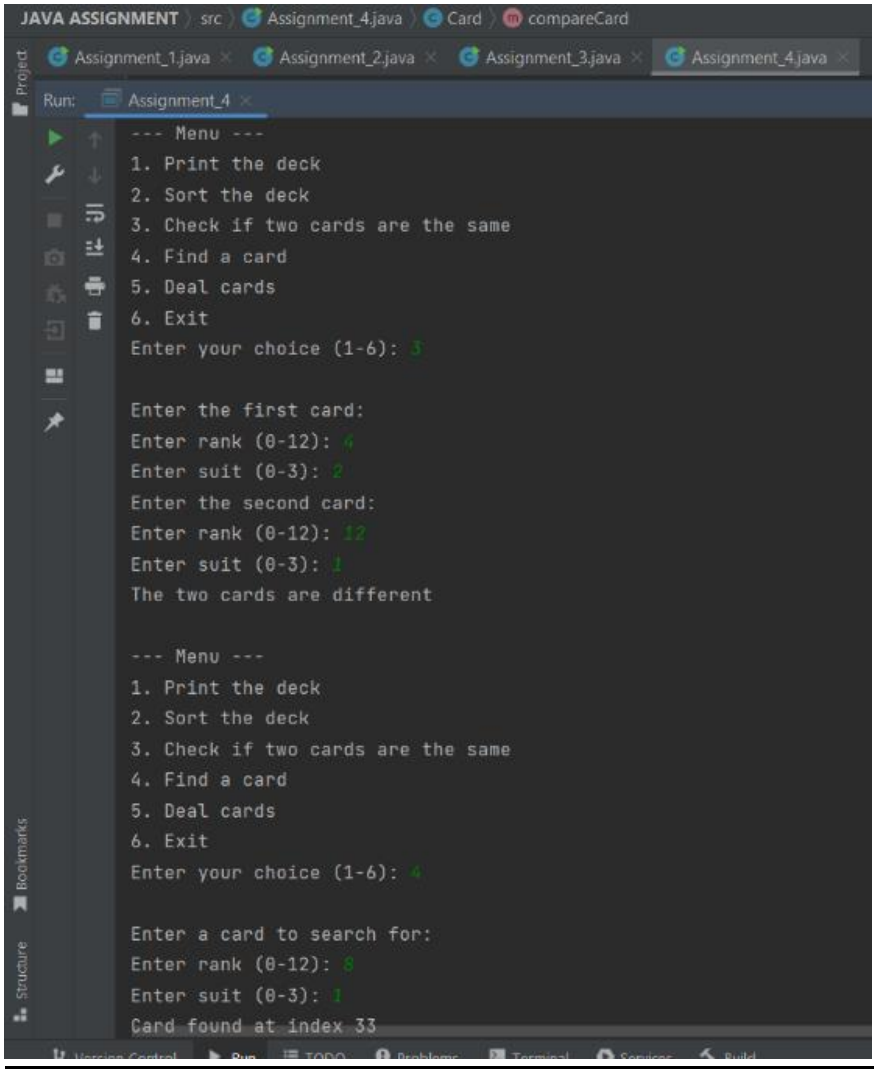

JAVA ASSIGNMENT
src
Assignment_4.java
Card
compareCard
Assignment_1.java
Assignment_2.java
Assignment_3.java
Assignment_4.java
Run:
Assignment_4
6 of Hearts
7 of Hearts
8 of Hearts
9 of Hearts
10 of Hearts
Jack of Hearts
Queen of Hearts
King of Hearts
Ace of Diamonds
2 of Diamonds
3 of Diamonds
4 of 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

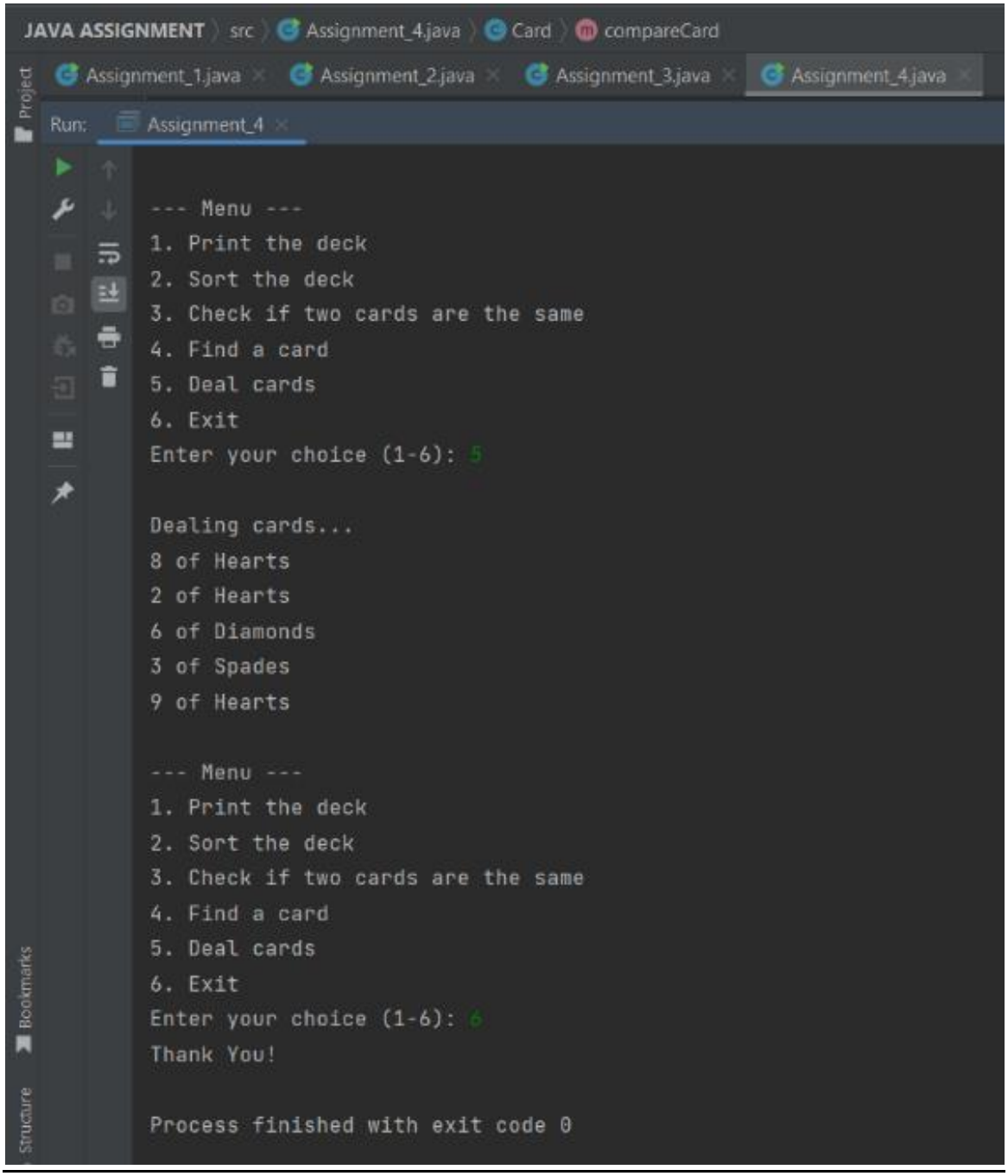

JAVA ASSIGNMENT
src
Assignment_4.java
Card
compareCard
Assignment_1.java
Assignment_2.java
Assignment_3.java
Assignment_4.java
Run:
Assignment_4
7 of Clubs
8 of Clubs
9 of Clubs
10 of Clubs
Jack of Clubs
Queen of Clubs
King of Clubs


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 Assignment_2.java Assignment_3.java Assignment_4.java
Run: Assignment_4
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 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): 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



GITHUB LINK :

https://github.com/anvesha31/Java_assignments/blob/634f675d715011fb9e73686217ca64655c947509/Assignment%204