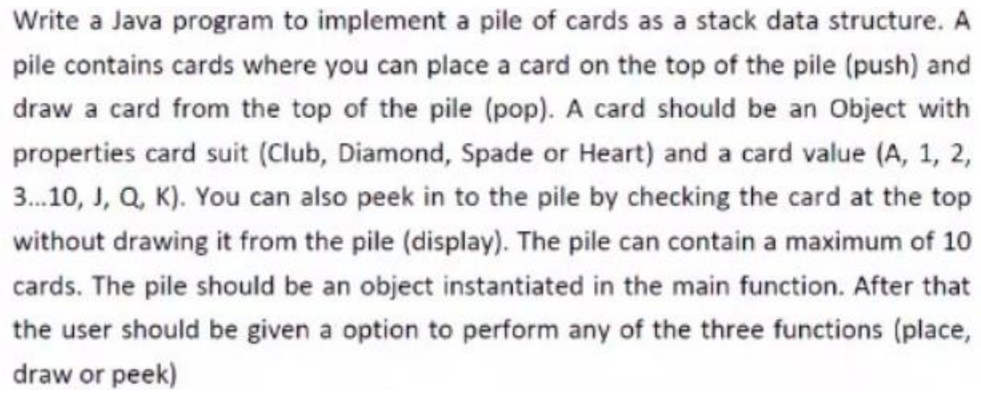
OOAD Lab Week 1

Date – 24 – Jan - 2022

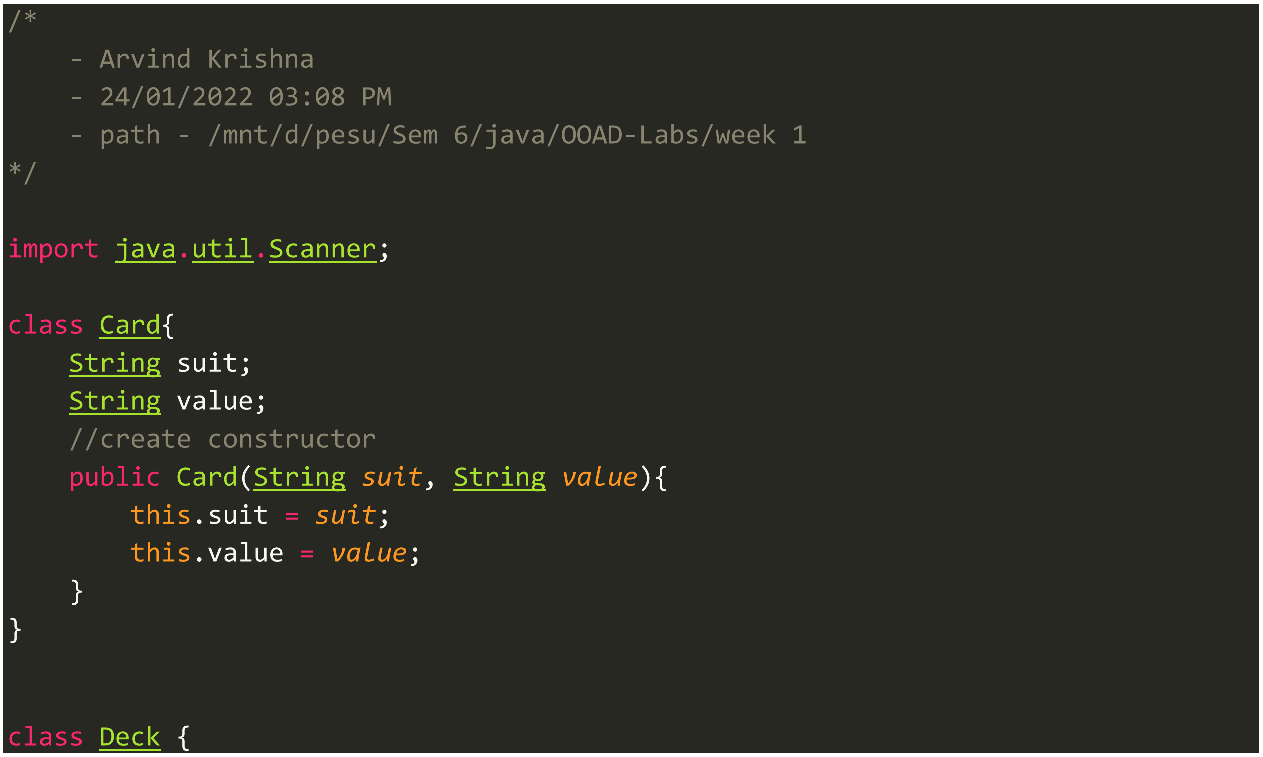
SRN – PES1UG19CS090	Name – Arvind Krishna	Sem – 6 Sec - B

Question:

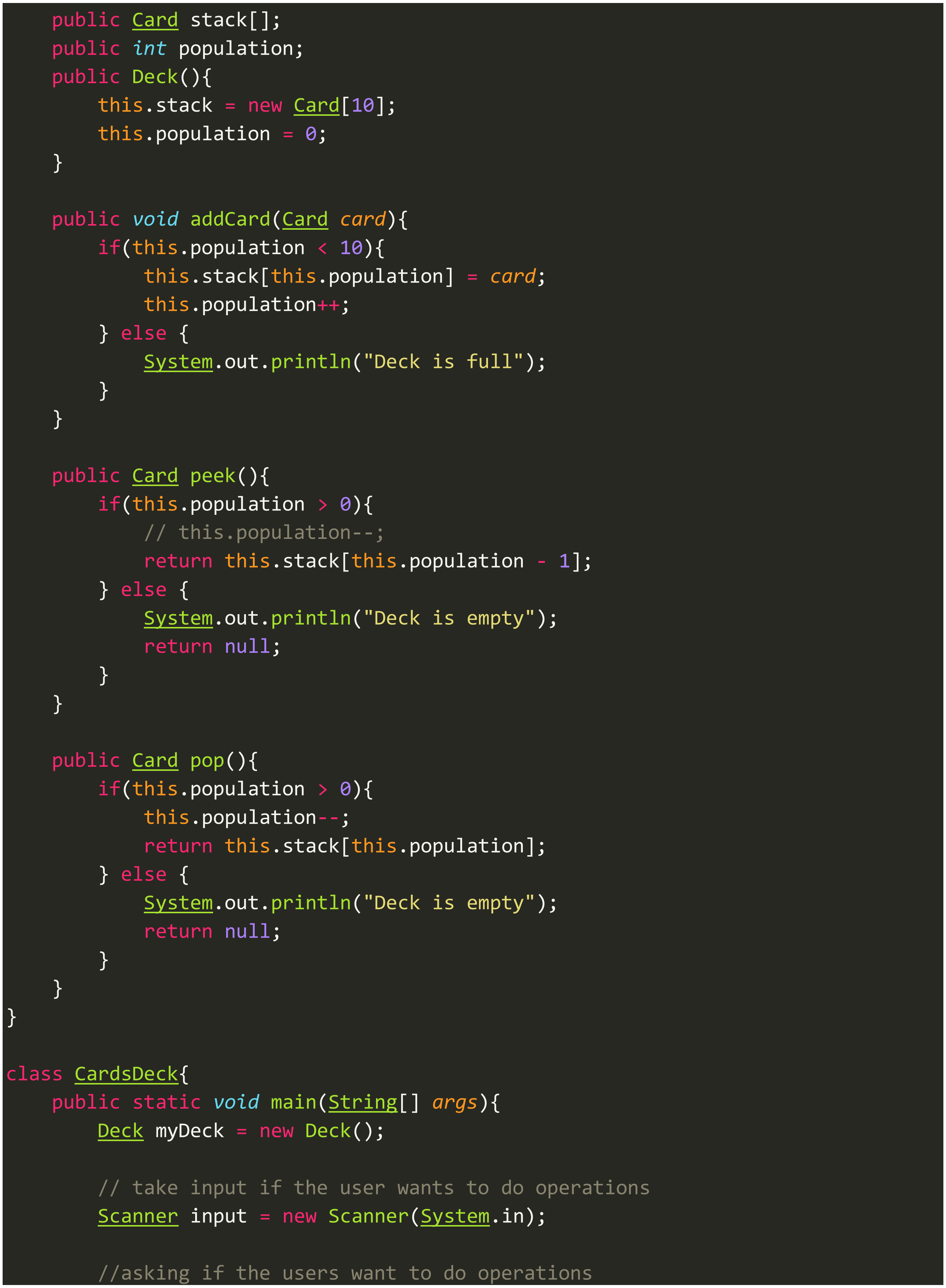


Write a Java program to implement a pile of cards as a stack data structure. A pile contains cards where you can place a card on the top of the pile (push) and draw a card from the top of the pile (pop). A card should be an Object with properties card suit (Club, Diamond, Spade or Heart) and a card value (A, 1, 2, 3...10, J, Q, K). You can also peek in to the pile by checking the card at the top without drawing it from the pile (display). The pile can contain a maximum of 10 cards. The pile should be an object instantiated in the main function. After that the user should be given a option to perform any of the three functions (place, draw or peek)

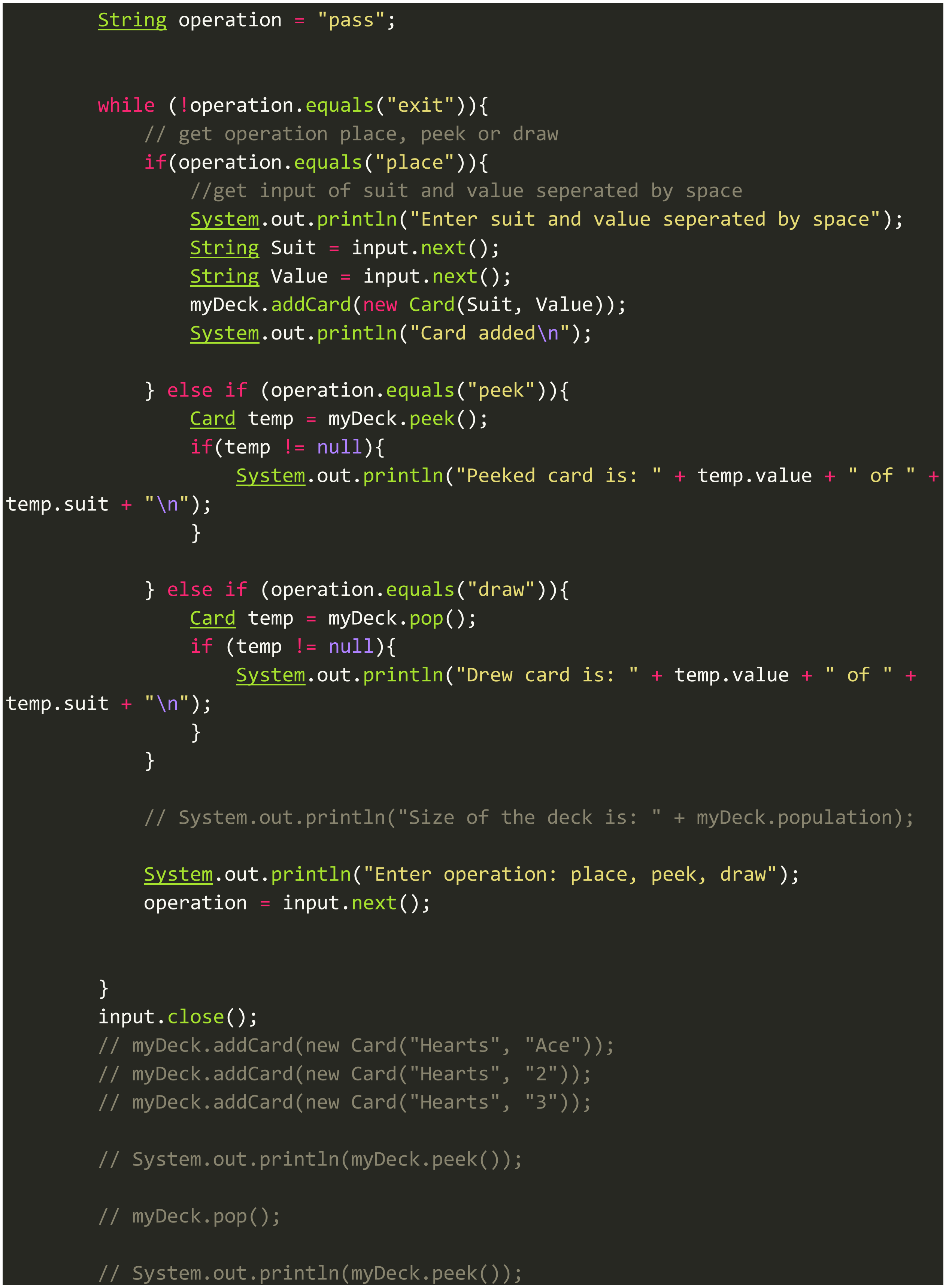
Code:



```java
/*
    - Arvind Krishna
    - 24/01/2022 03:08 PM
    - path - /mnt/d/pesu/Sem 6/java/OOAD-Labs/week 1
*/

import java.util.Scanner;

class Card{
    String suit;
    String value;
    //create constructor
    public Card(String suit, String value){
        this.suit = suit;
        this.value = value;
    }
}


class Deck {
```



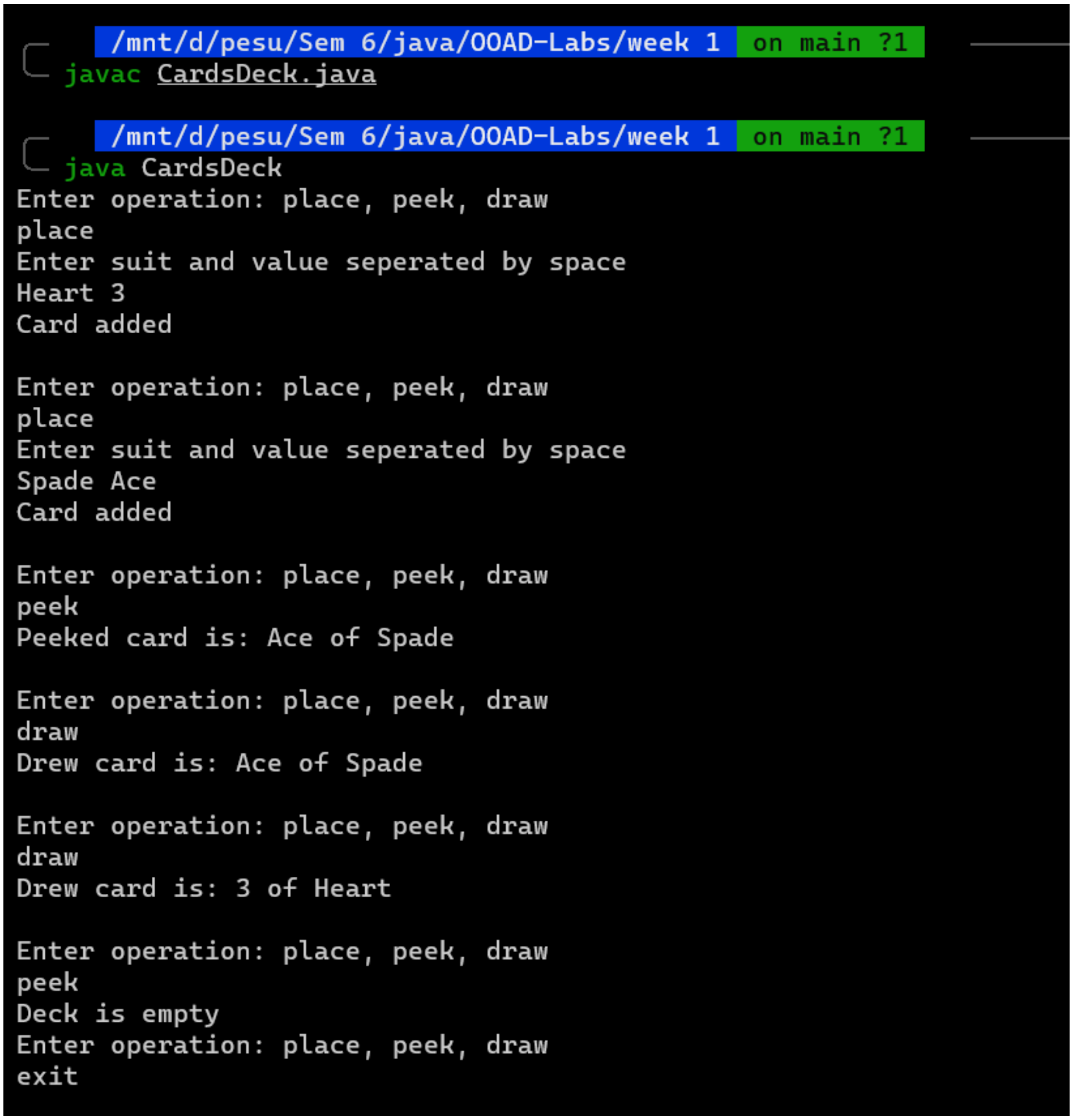
```
    public Card stack[];
    public int population;
    public Deck(){
        this.stack = new Card[10];
        this.population = 0;
    }

    public void addCard(Card card){
        if(this.population < 10){
            this.stack[this.population] = card;
            this.population++;
        } else {
            System.out.println("Deck is full");
        }
    }

    public Card peek(){
        if(this.population > 0){
            // this.population--;
            return this.stack[this.population - 1];
        } else {
            System.out.println("Deck is empty");
            return null;
        }
    }

    public Card pop(){
        if(this.population > 0){
            this.population--;
            return this.stack[this.population];
        } else {
            System.out.println("Deck is empty");
            return null;
        }
    }
}

class CardsDeck{
    public static void main(String[] args){
        Deck myDeck = new Deck();

        // take input if the user wants to do operations
        Scanner input = new Scanner(System.in);

        //asking if the users want to do operations
```



```
        String operation = "pass";


        while (!operation.equals("exit")){
            // get operation place, peek or draw
            if(operation.equals("place")){
                //get input of suit and value seperated by space
                System.out.println("Enter suit and value seperated by space");
                String Suit = input.next();
                String Value = input.next();
                myDeck.addCard(new Card(Suit, Value));
                System.out.println("Card added\n");

            } else if (operation.equals("peek")){
                Card temp = myDeck.peek();
                if(temp != null){
                    System.out.println("Peeked card is: " + temp.value + " of " +
temp.suit + "\n");
                }

            } else if (operation.equals("draw")){
                Card temp = myDeck.pop();
                if (temp != null){
                    System.out.println("Drew card is: " + temp.value + " of " +
temp.suit + "\n");
                }
            }

            // System.out.println("Size of the deck is: " + myDeck.population);

            System.out.println("Enter operation: place, peek, draw");
            operation = input.next();


        }
        input.close();
        // myDeck.addCard(new Card("Hearts", "Ace"));
        // myDeck.addCard(new Card("Hearts", "2"));
        // myDeck.addCard(new Card("Hearts", "3"));

        // System.out.println(myDeck.peek());

        // myDeck.pop();

        // System.out.println(myDeck.peek());
```

```
    }
}
```

Output Screenshots:



```
    /mnt/d/pesu/Sem 6/java/OOAD-Labs/week 1  on main ?1
   javac CardsDeck.java

    /mnt/d/pesu/Sem 6/java/OOAD-Labs/week 1  on main ?1
   java CardsDeck
Enter operation: place, peek, draw
place
Enter suit and value seperated by space
Heart 3
Card added

Enter operation: place, peek, draw
place
Enter suit and value seperated by space
Spade Ace
Card added

Enter operation: place, peek, draw
peek
Peeked card is: Ace of Spade

Enter operation: place, peek, draw
draw
Drew card is: Ace of Spade

Enter operation: place, peek, draw
draw
Drew card is: 3 of Heart

Enter operation: place, peek, draw
peek
Deck is empty
Enter operation: place, peek, draw
exit
```