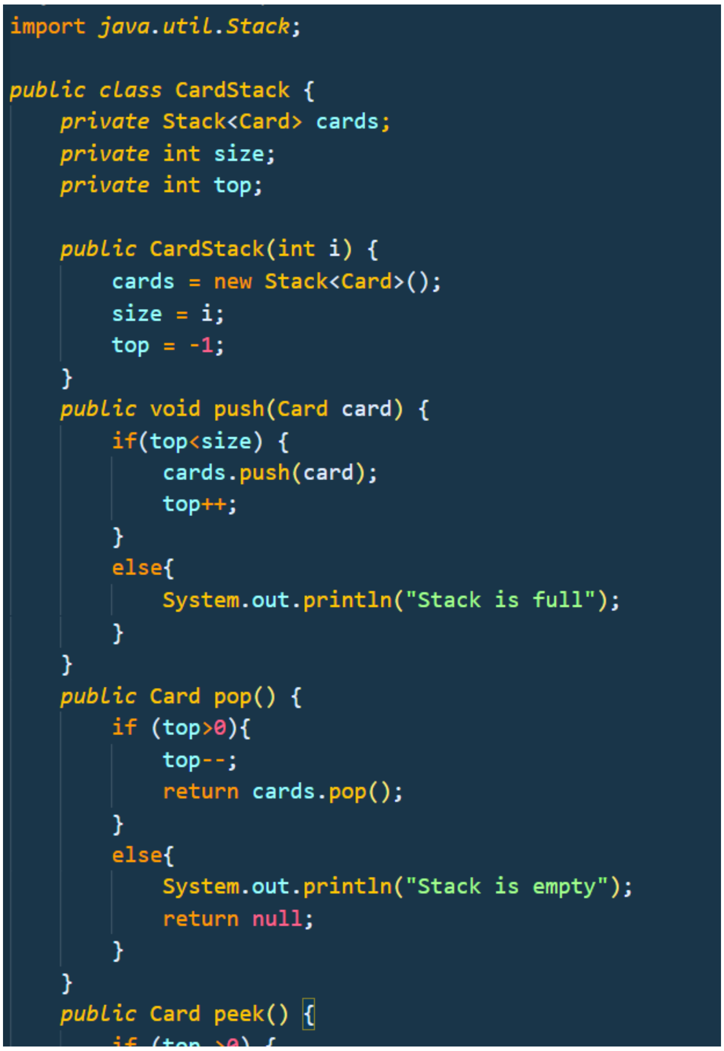
Name: Venkata Krishnarjun Vuppala
SRN: PES2UG19CS451
Subject: OOAD

Semester: 6
Section: G

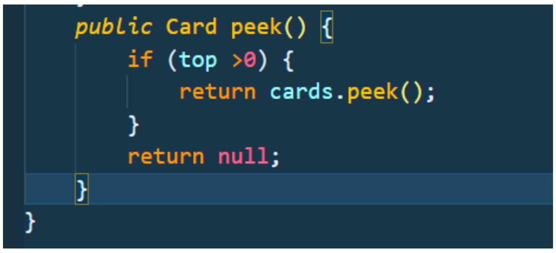
Assignment 1

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 onthe 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)

CardStack.java

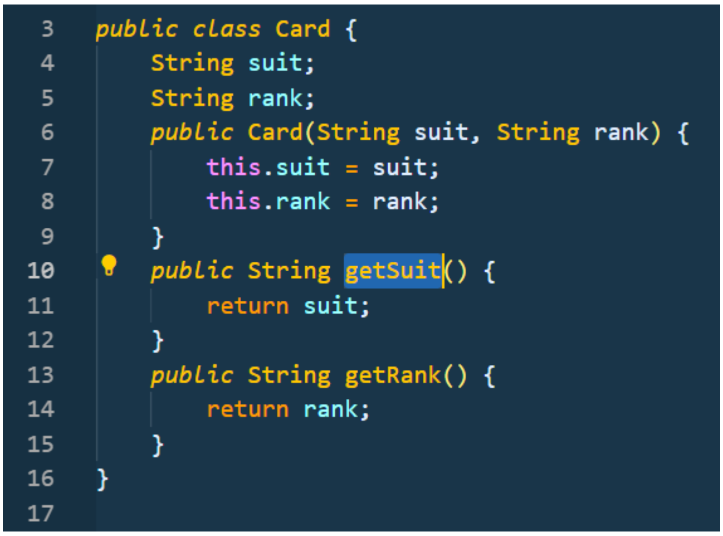


```java
import java.util.Stack;

public class CardStack {
    private Stack<Card> cards;
    private int size;
    private int top;

    public CardStack(int i) {
        cards = new Stack<Card>();
        size = i;
        top = -1;
    }
    public void push(Card card) {
        if(top<size) {
            cards.push(card);
            top++;
        }
        else{
            System.out.println("Stack is full");
        }
    }
    public Card pop() {
        if (top>0){
            top--;
            return cards.pop();
        }
        else{
            System.out.println("Stack is empty");
            return null;
        }
    }
    public Card peek() {
        if (top >0) {
```



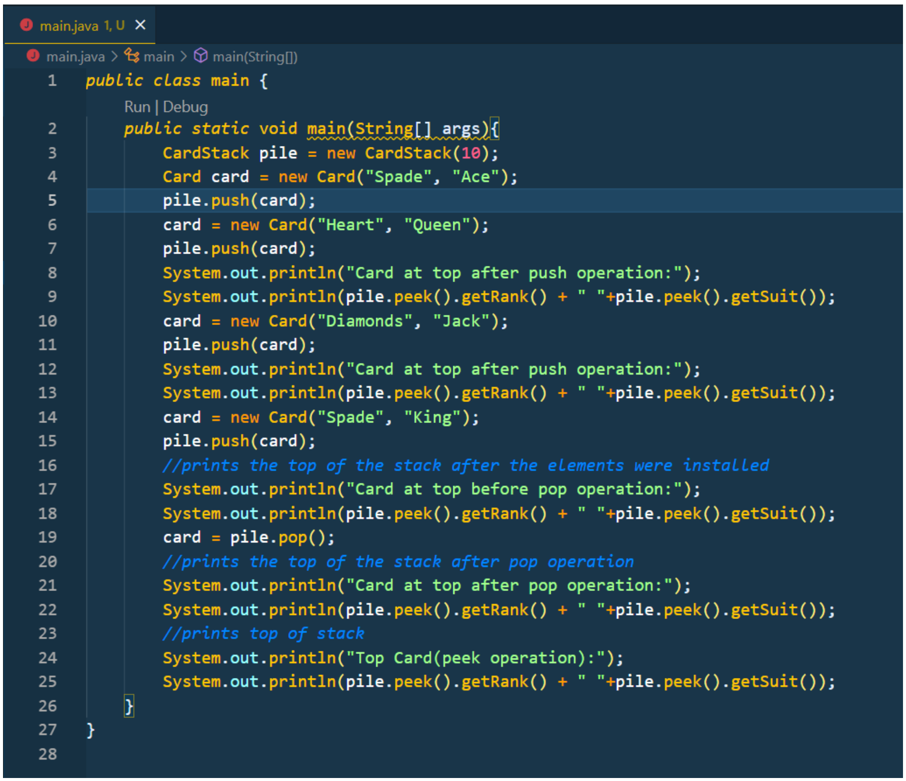
```java
    public Card peek() {
        if (top >0) {
            return cards.peek();
        }
        return null;
    }
}
```

Card.java



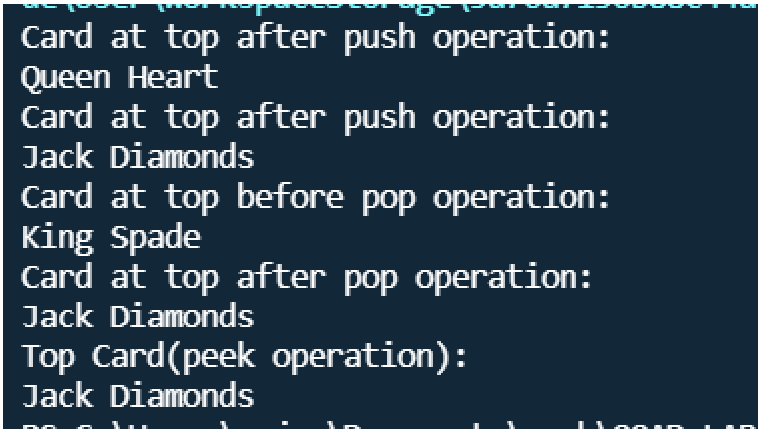
```java
public class Card {
    String suit;
    String rank;
    public Card(String suit, String rank) {
        this.suit = suit;
        this.rank = rank;
    }
    public String getSuit() {
        return suit;
    }
    public String getRank() {
        return rank;
    }
}
```

Main.java



```java
public class main {
    public static void main(String[] args){
        CardStack pile = new CardStack(10);
        Card card = new Card("Spade", "Ace");
        pile.push(card);
        card = new Card("Heart", "Queen");
        pile.push(card);
        System.out.println("Card at top after push operation:");
        System.out.println(pile.peek().getRank() + " "+pile.peek().getSuit());
        card = new Card("Diamonds", "Jack");
        pile.push(card);
        System.out.println("Card at top after push operation:");
        System.out.println(pile.peek().getRank() + " "+pile.peek().getSuit());
        card = new Card("Spade", "King");
        pile.push(card);
        //prints the top of the stack after the elements were installed
        System.out.println("Card at top before pop operation:");
        System.out.println(pile.peek().getRank() + " "+pile.peek().getSuit());
        card = pile.pop();
        //prints the top of the stack after pop operation
        System.out.println("Card at top after pop operation:");
        System.out.println(pile.peek().getRank() + " "+pile.peek().getSuit());
        //prints top of stack
        System.out.println("Top Card(peek operation):");
        System.out.println(pile.peek().getRank() + " "+pile.peek().getSuit());
    }
}
```

Output



```
Card at top after push operation:
Queen Heart
Card at top after push operation:
Jack Diamonds
Card at top before pop operation:
King Spade
Card at top after pop operation:
Jack Diamonds
Top Card(peek operation):
Jack Diamonds
```