Stack Ki Tafseelat:

1. Stack Kya Hai?

- Stack ek data structure hai jo Last In, First Out (LIFO) order mein kaam karta hai.
- Matlab, jo cheez sab se aakhir mein daali gayi ho wo sab se pehle bahar aati hai.

2. Stack Operations:

- **Push:** Naya element stack mein daalne ke liye hota hai. Jaise aap books ek dusre ke upar rakh rahe hote hain.
- **Pop:** Stack se upar wala element nikalne ke liye hota hai. Jaise aap ek ek karke books niche se nikalte hain.
- **Peek:** Stack ka top element dekhne ke liye hota hai. Yeh bina element ko stack se nikale, sirf dekhta hai.
- **isEmpty:** Yeh check karta hai ke stack khaali hai ya nahi.
- **isFull:** Yeh check karta hai ke stack full hai ya nahi.

3. Stack Implementation (Java Code):

```
import java.util.EmptyStackException;

class Stack {
    private int maxSize;
    private int[] stackArray;
    private int top;

    // Constructor
    public Stack(int maxSize) {
        this.maxSize = maxSize;
        this.stackArray = new int[maxSize];
        this.top = -1;
    }

    // Push operation: element ko stack mein daalne ke liye
    public void push(int element) {
        if (isFull()) {
            System.out.println("Stack full, cannot push element.");
            return;
        }
        stackArray[++top] = element;
        System.out.println(element + " pushed to stack.");
    }

    // Pop operation: stack se element nikalne ke liye
    public int pop() {
```

```
        if (isEmpty()) {
            throw new EmptyStackException();
        }
        int poppedElement = stackArray[top--];
        System.out.println(poppedElement + " popped from stack.");
        return poppedElement;
    }

    // Peek operation: stack ka top element dekhne ke liye
    public int peek() {
        if (isEmpty()) {
            throw new EmptyStackException();
        }
        return stackArray[top];
    }

    // isEmpty operation: stack khaali hai ya nahi
    public boolean isEmpty() {
        return (top == -1);
    }

    // isFull operation: stack full hai ya nahi
    public boolean isFull() {
        return (top == maxSize - 1);
    }
}

public class Main {
    public static void main(String[] args) {
        Stack stack = new Stack(5);

        // Stack operations
        stack.push(10);
        stack.push(20);
        stack.push(30);
        stack.push(40);
        stack.push(50);

        System.out.println("Top element of stack: " + stack.peek());

        stack.pop();
        stack.pop();
        stack.pop();
        stack.pop();
        stack.pop(); // Trying to pop from empty stack
    }
}
```

Code Explanation (Code Ki Tafseelat):

1. **Class Stack:**
    - **Fields:**
        - `maxSize`: Stack ka maximum size, kitne elements ko store kar sakta hai.
        - `stackArray`: Array jo stack ko store karega.
        - `top`: Stack ka top element ka index.
    - **Constructor:**
        - `public Stack(int maxSize)`: Constructor jisme stack ki size set hoti hai, aur array banai jati hai.
    - **Methods:**
        - `public void push(int element)`: Naya element stack mein daalne ke liye.
        - `public int pop()`: Stack se upar wala element nikalne ke liye.
        - `public int peek()`: Stack ka top element dekhne ke liye.
        - `public boolean isEmpty()`: Check karta hai ke stack khaali hai ya nahi.
        - `public boolean isFull()`: Check karta hai ke stack full hai ya nahi.
2. **Class Main:**
    - **Main Method:**
        - `public static void main(String[] args)`: Yahan stack create kiya jata hai aur us par operations perform kiye jate hain.
    - **Stack Operations:**
        - `stack.push(10);`: 10 ko stack mein daala gaya.
        - `stack.push(20);`: 20 ko stack mein daala gaya.
        - `stack.push(30);`: 30 ko stack mein daala gaya.
        - `stack.push(40);`: 40 ko stack mein daala gaya.
        - `stack.push(50);`: 50 ko stack mein daala gaya.
        - `System.out.println("Top element of stack: " + stack.peek());`: Stack ka top element print kiya gaya.
        - `stack.pop();`: Stack se ek element nikala gaya.

- `stack.pop();`: Stack se ek aur element nikala gaya.
- `stack.pop();`: Stack se ek aur element nikala gaya.
- `stack.pop();`: Stack se ek aur element nikala gaya.
- `stack.pop();`: Trying to pop from an empty stack, isse error aayega.

Note: Code mein `push`, `pop`, aur `peek` operations ka istemal stack ke basic operations ko demonstrate karne ke liye kiya gaya hai. Stack ki size 5 hai, isliye 6th element push karne par error aayega.

4. Stack Ka Istemal:

- Memory management mein, function calls ko track karne ke liye stack ka istemal hota hai. Jab ek function call hoti hai, toh uska data stack mein store hota hai aur jab woh call complete hoti hai, tab us data ko stack se nikala jata hai.

5. Stack Ki Misal:

- Jaise aap ek plate ki stack banate hain. Aap sabse upar ek plate rakhenge, jab aapko ek aur plate chahiye hoti hai toh aap usko sabse upar rakhenge. Jab aapko kisi plate ki zarurat hoti hai toh aap sabse upar wali plate nikalenge.

Stack ka concept bahut sari programming situations mein istemal hota hai, aur yeh ek important data structure hai jo bahut se algorithms mein bhi istemal hota hai.