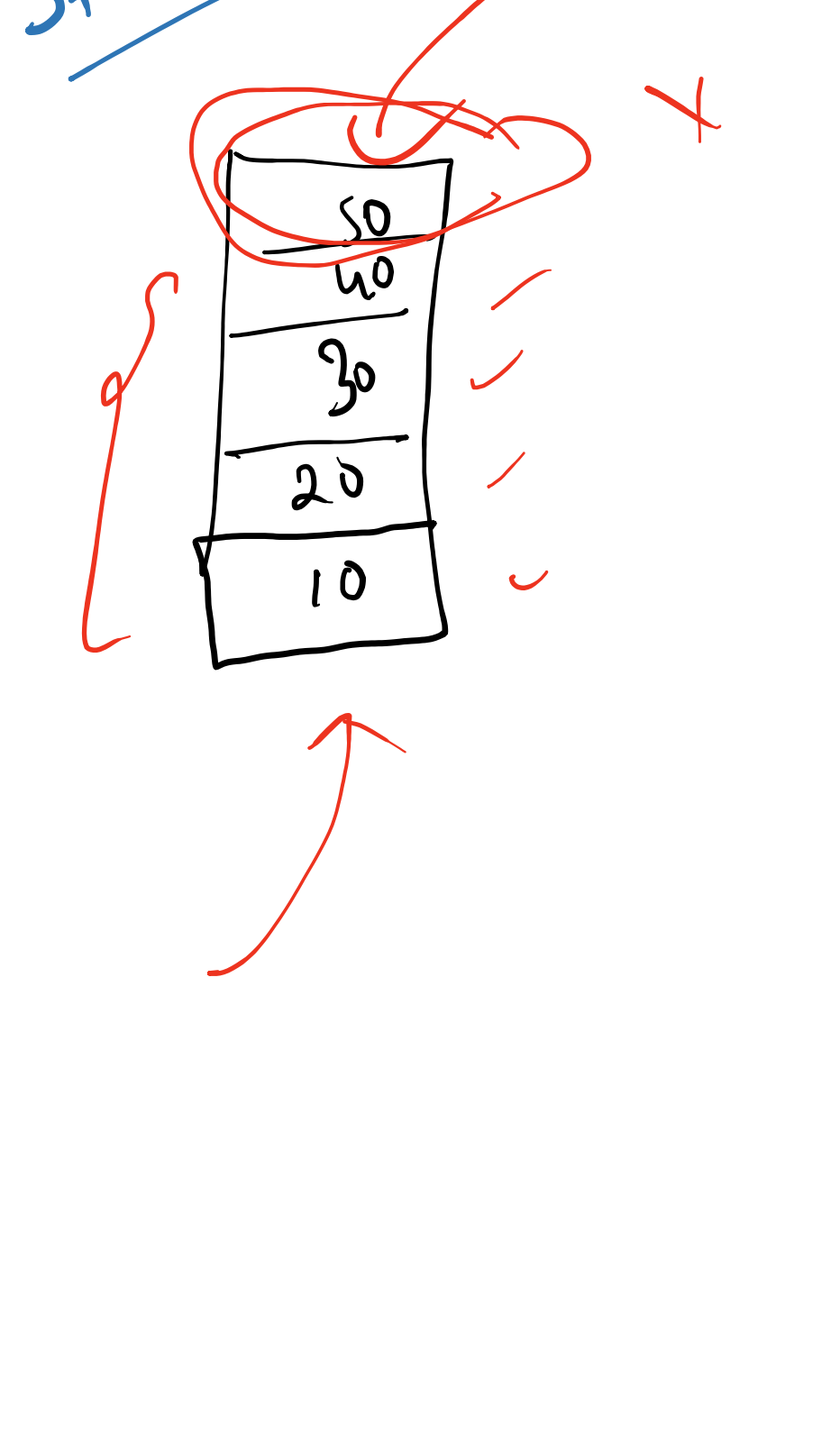
Lec-23

17/05/25 9:54 AM

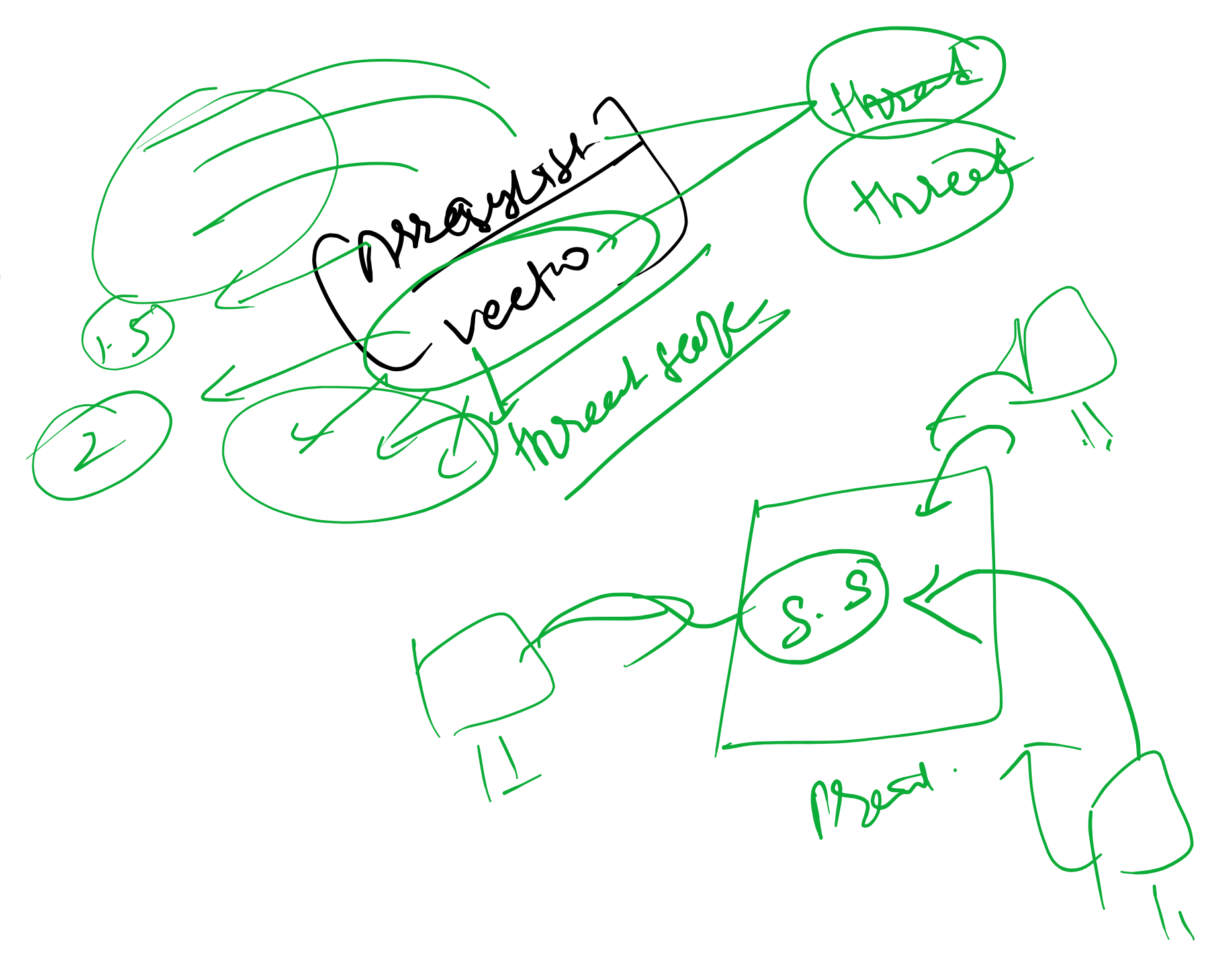
Stack

LIFO



Java ArrayList vector

Vector



(i) Push
(ii) pop
(iii) peek
(iv) size
(v) Empty()
(vi) display → for each

N=5

St.push(10)

```
public boolean isfull() {
}
public void push(int item) {
}
```

idx = arr.length-1

idx++
arr[idx]=item

St.push(10)
St.push(20)
St.push(30)
St.push(40)
St.push(50)

FIFO

Queue

Class

Interface

Linear Queue

Circular Queue

Dequeue

Stack class

Enqueue ← add
dequeue ← remove
get front ← get
size
isEmpty
display

front

rear

add

```
public boolean isEmpty() {
    return size==0;
}
public boolean isfull() {
    return size==arr.length;
}
public int  size() {
    return size;
}
public void Enqueue(int item) {
}
```

arr[size]=item
size++

q.Enq(10)
q.Enq(20)
q.Enq(30)
q.Enq(40)
q.Enq(50)

```
public boolean isEmpty() {
    return size == 0;
}

public boolean isfull() {
    return size == arr.length;
}

public int size() {
    return size;
}

public void Enqueue(int item) throws Exception {
    if (isfull()) {
        throw new Exception("Bklol Queue full hogya ");
    }
    arr[size] = item;
    size++;
}
public int dequeue() {
}
```

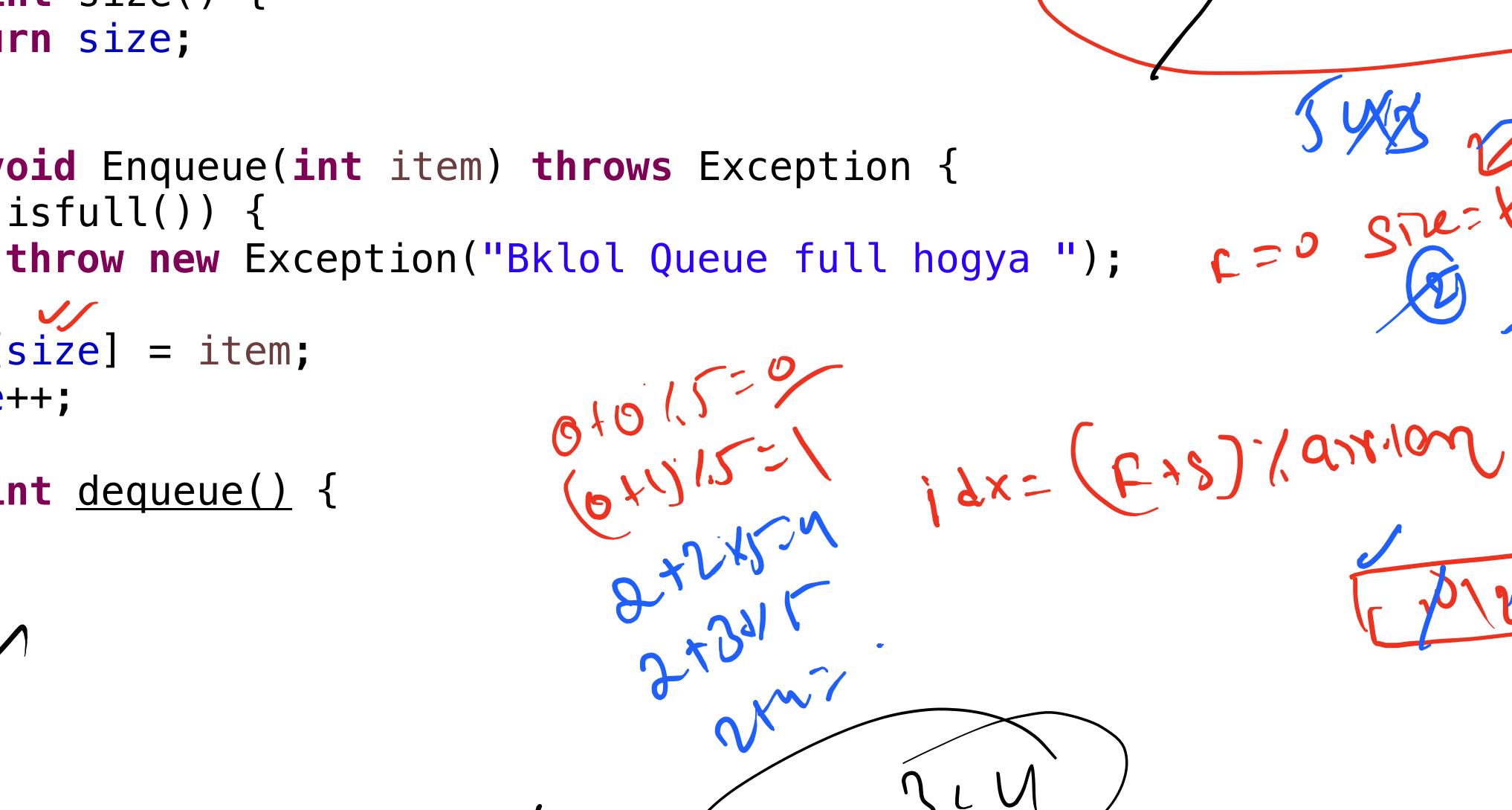
v = arr[front]
front++
size--

idx = (f+s) % arr.len

for (i=0; i < size; i++) {
int idx = (f+i) % arr.length

size = 4
front = 2

30 40 50 60



```
public static void main(String[] args) {
        // TODO Auto-generated method stub
        Stack<Integer> st = new Stack<>();
        st.push(10);
        st.push(20);
        st.push(30);
        st.push(40);
        st.push(50);
        System.out.println(st);
        addlast(st, 7);
        System.out.println(st);
    }

    private static void addlast(Stack<Integer> st, int item) {
        // TODO Auto-generated method stub
        if (st.isEmpty()) {
            st.push(item);
            return;
        }
        int x = st.pop();
        addlast(st, item);
        st.push(x);
    }
```