

UNIT-II(T201 DATA STRUCTURES) ONE

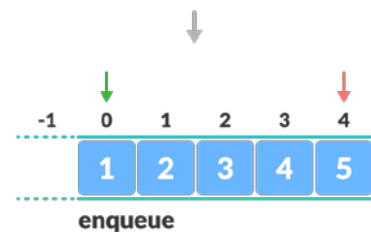
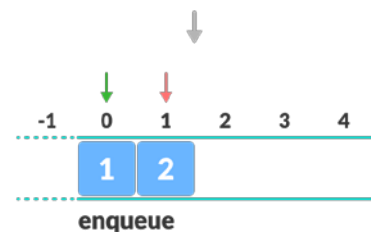
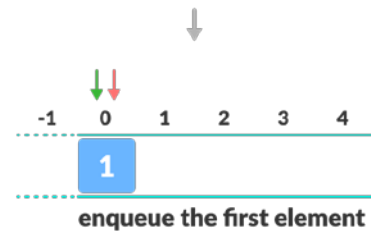
Queue: Representation of Queue, Operations on Queue, Circular Queue, Priority Queue, Array representation of Priority Queue, Double Ended Queue, Applications of Queue

A queue is an object (an abstract data structure - ADT) that allows the following operations:

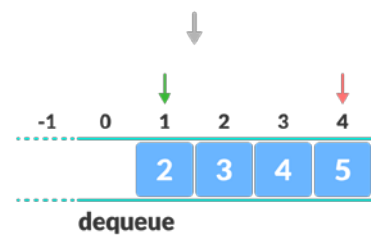
- **Enqueue:** Add an element to the end of the queue
- **Dequeue:** Remove an element from the front of the queue
- **IsEmpty:** Check if the queue is empty
- **IsFull:** Check if the queue is full
- **Peek:** Get the value of the front of the queue without removing it



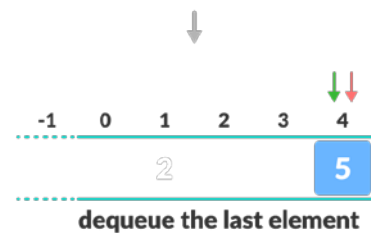
```
import java.util.Scanner;
public class aqueue {
    int aqueueLength = 3;
    int items[] = new int[aqueueLength];
    int front = -1;
    int rear = -1;
    boolean isFull(){
        if(rear == aqueueLength - 1){
            return true;
        } else {
            return false;
        }
    }
    boolean isEmpty(){
        if(front == -1 && rear == -1){
            return true;
        } else {
            return false;
        }
    }
}
```



```
void enaqueue(int itemValue) {
    if(isFull()){
        System.out.println("aqueue is full");
    } else if(front == -1 && rear == -1){
        front = rear = 0;
        items[rear] = itemValue;
    } else{
        rear++;
        items[rear] = itemValue;
    }
}
```



```
int deaqueue(){
    if(isEmpty()){
        System.out.println("aqueue is empty. Nothing to deaqueue");
        return 0;
    } else if (front == rear){
        front = rear = -1;
        return 0;
    }
}
```



```

    } else {
        int x=items[front];
        front++;
        return x;
    }
}
void display(){
    int i;
    if(isEmpty()){
        System.out.println("aqueue is empty");
    } else {
        for(i = front; i <= rear; i++){
            System.out.println(items[i]);
        }
    }
}
void peep(){
    System.out.println("Front value is: " + items[front]);
}
public static void main(String[] args) {
    aqueue q = new aqueue();
    Scanner sc=new Scanner(System.in);
    while(true){
        System.out.println("Menu---> 1.INSERT 2. DELETE 3. PEEK 4. DISPLAY 5.EXIT");
        System.out.println("Enter Choice :");
        int ch=sc.nextInt();
        switch(ch){
            case 1:
                System.out.println("Enter Element to Insert :");
                int Ele=sc.nextInt();
                q.enaqueue(Ele);
                break;
            case 2:
                System.out.println(q.deaqueue() + " Deleted from QQQ");
                break;
            case 3:
                q.peep();
                break;
            case 4:
                System.out.println("Elements present in Queue :");
                q.display();
                break;
            case 5:
                System.exit(0);
            default:
                System.out.println("Invalid Choice");
        }
    }
}
}
}

```