Data Structure and Algorithms

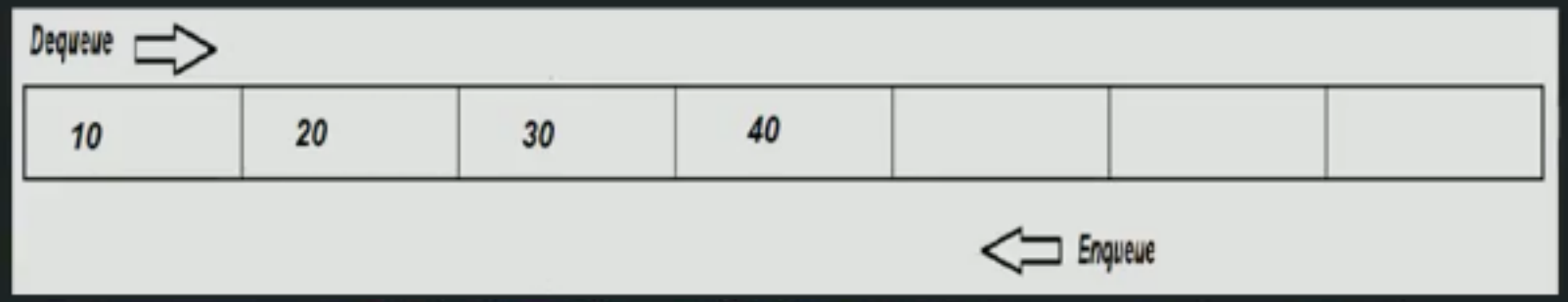
Session-11

Dr. Subhra Rani Patra
SCOPE, VIT Chennai

What is a Queue:

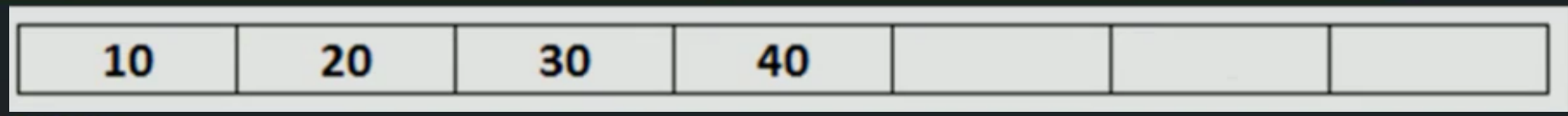
✓ Observations from above Picture:

- ✓ New addition of members happens at end of the queue.
- ✓ First person in the queue is the first to get out from queue.
- ✓ follows FIFO (First in First Out) method


Dequeue
10
20
30
40
Enqueue


COFFEE
ORDER HERE
GRAVITY

Implementation Options of Queue:



- ✓ **Array:**
  - ✓ Linear Queue
  - ✓ Circular Queue

Common operations in Queue:

- ✓ createQueue()
- ✓ enQueue()
- ✓ deQueue()
- ✓ peekInQueue()
- ✓ isEmpty()
- ✓ isFull()
- ✓ deleteQueue()

Creation of Linear Queue (Array Implementation):

```
createQueue(size)

    create a blank array of 'size'

    initialize topOfQueue, beginningOfQueue to -1
```

Enqueue operation of Linear Queue(Array Implementation):

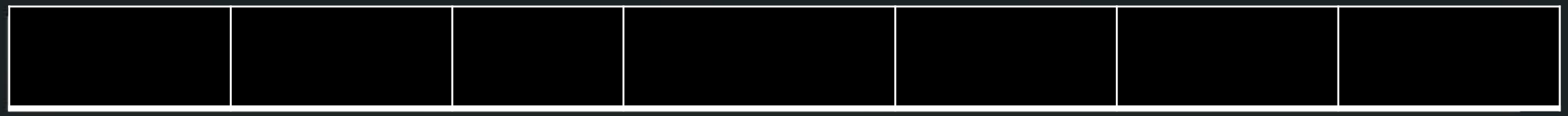
```
enQueue(Value):
    if Queue is full
        return error message
    else
        arr [topOfQueue + 1 ] = Value
        topOfQueue ++
```

Dequeue operation of Linear Queue(Array Implementation):

10	20	30	40	50		

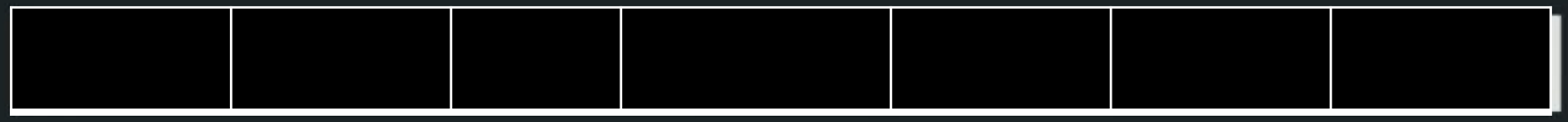
```
deQueue():
    if queue is empty
        return error message
    else
        print arr [beginningOfQueue]
        beginningOfQueue ++
        if (beginningOfQueue > endOfQueue) //If last element in the Queue is Dequeued
            beginningOfQueue = topOfQueue = -1
```

Peek operation of Linear Queue(Array Implementation):



```
peek()
    if queue is empty
        return error message
    else
        print arr [beginningOfQueue]
```

IsEmpty operation of Linear Queue(Array Implementation):



```
IsQueueEmpty()
    if ((beginningOfQueue == -1)
        return true
    else
        return false
```

IsFull operation of Linear Queue(Array Implementation):

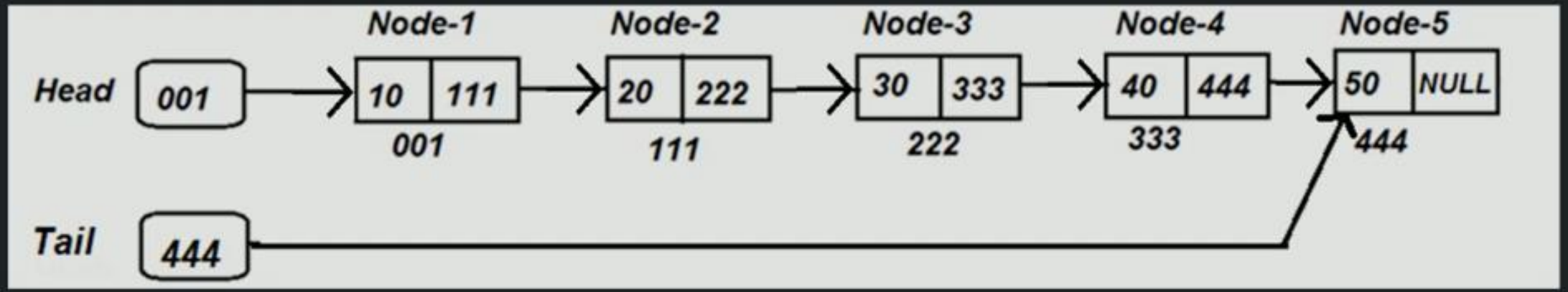
```
isQueueFull()
    if (topOfQueue == arr.length-1)
        return true
    else
        return false
```

Deleting a Linear Queue(Array Implementation):

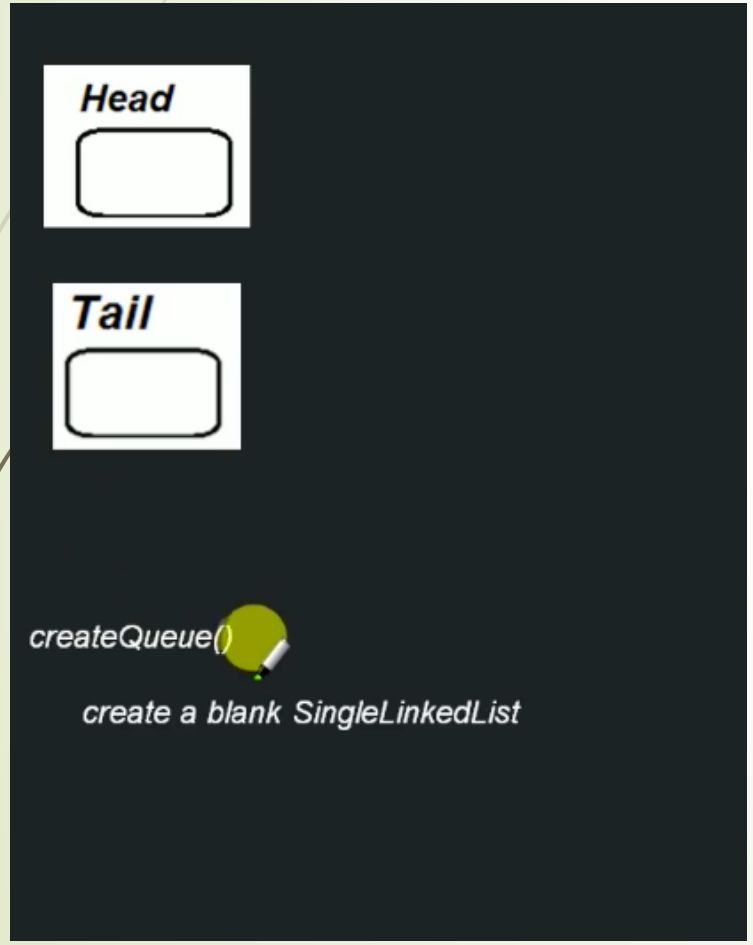
```
deleteQueue():

    array = null
```

Implementation Options of Queue:

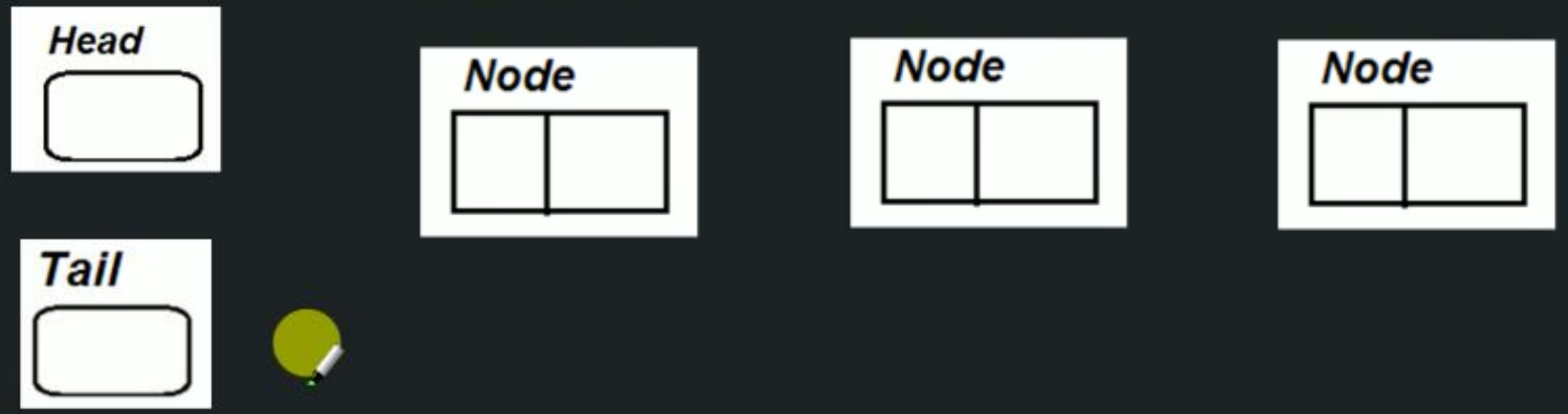
- ✓ **Linked List**
  - ✓ Linear Queue



Creation of Queue(Linked List Implementation):

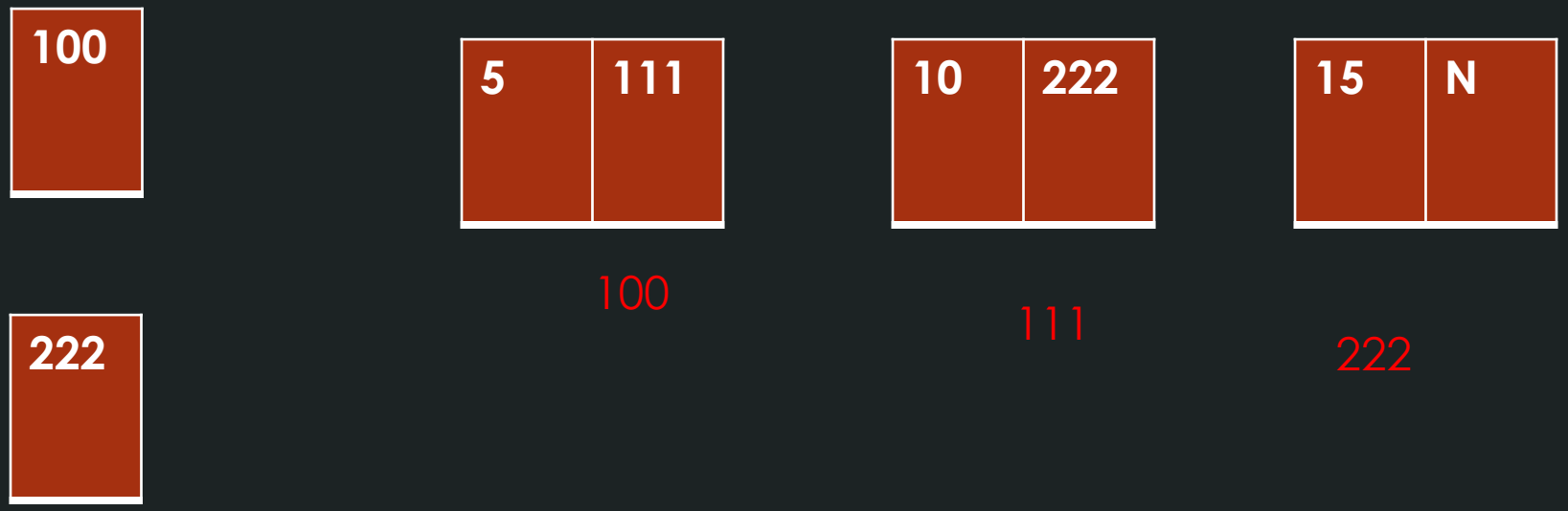


Enqueue operation of Queue(Linked List Implementation):




```
enQueue(nodeValue):
    create a node
    node.value = nodeValue
    node.next = null
    if tail equals null // if queue is empty
head =  tail = node
    else  //if queue is not empty
       tail.next = node
       tail = node
```

Dequeue operation of Queue(Linked List Implementation):



deQueue():

if header equals null

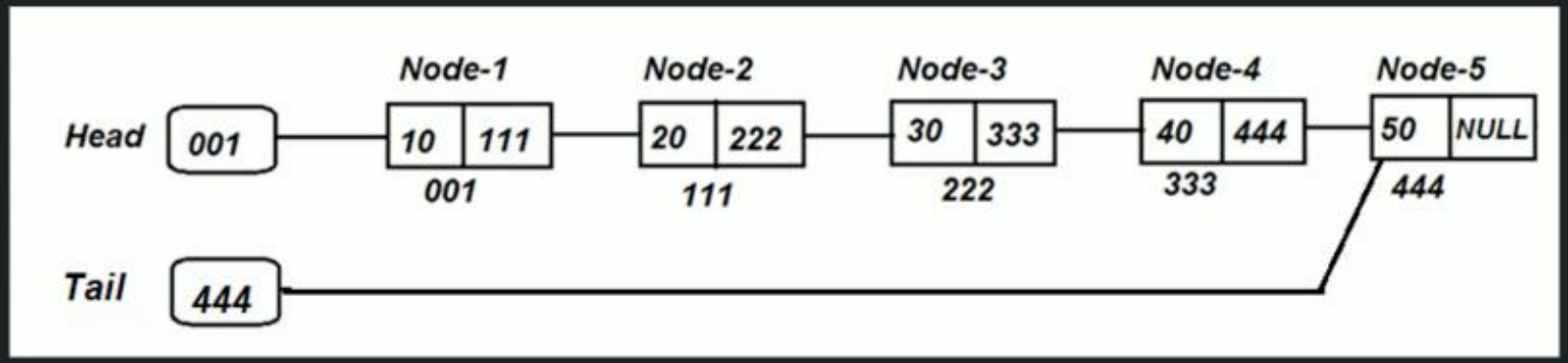
return error message

tmpNode = header

header = header.next

return tmpNode.value

Peek operation of Queue(Linked List Implementation):



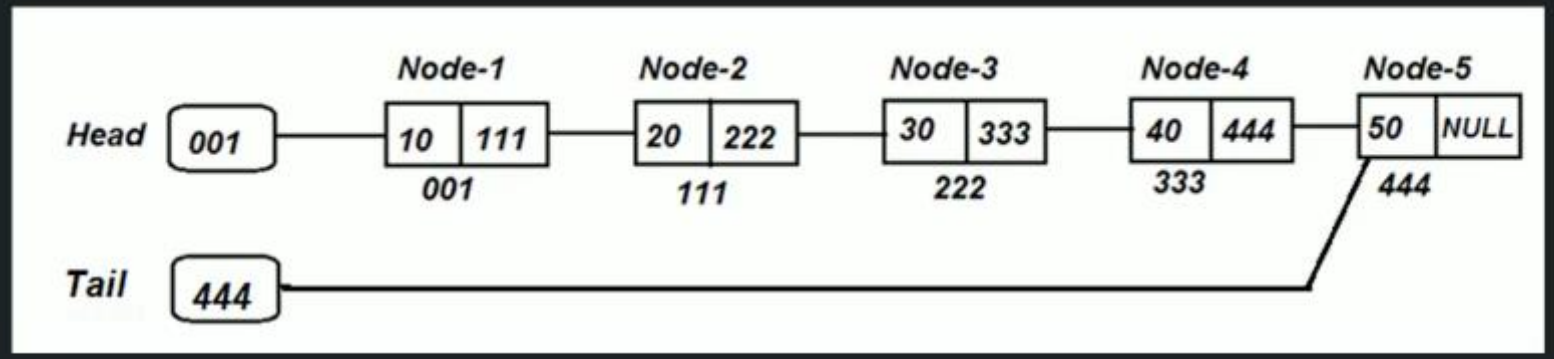
peek():

```
if header equals null
    return error
else
    return header.value
```

IsEmpty operation of Queue(Linked List Implementation):

```
isQueueEmpty():
    if (header equals null)
        return true
    else
        return false
```

Deleting a Queue(Linked List Implementation):



deleteQueue():

head = tail = null


Thank
you