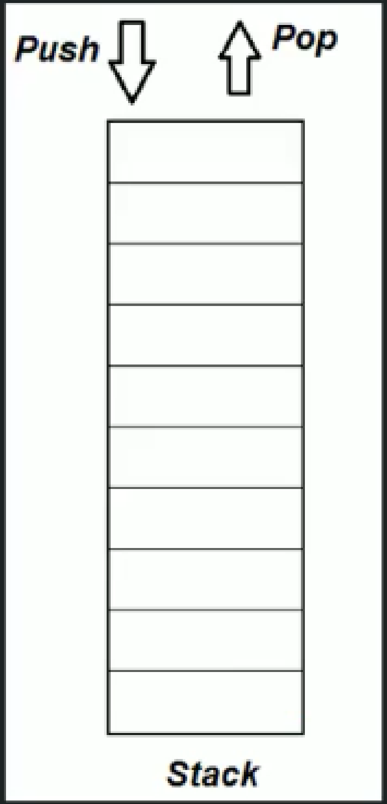
Data Structure and Algorithms

Session-6

Dr. Subhra Rani Patra
SCOPE, VIT Chennai

What is Stack ?

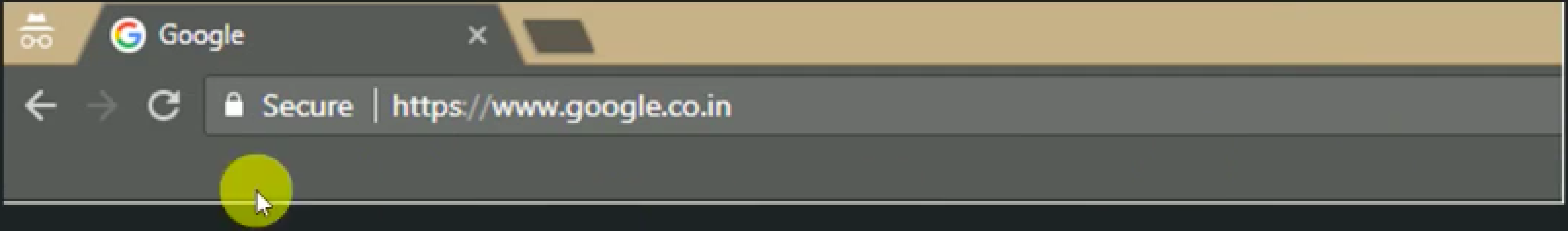


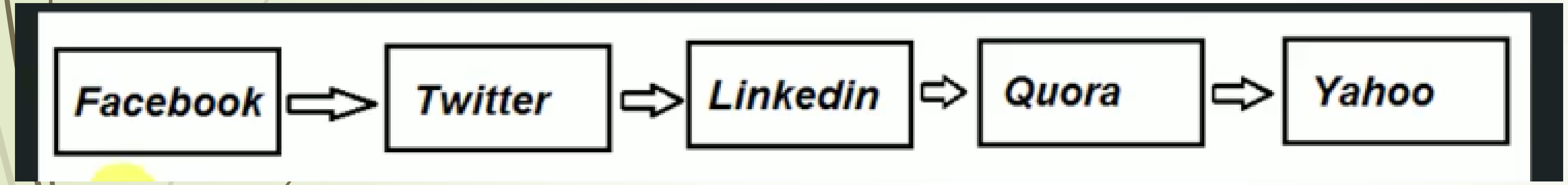
- ✓ Property of Stack:
  - ✓ follows LIFO (Last in First Out) method

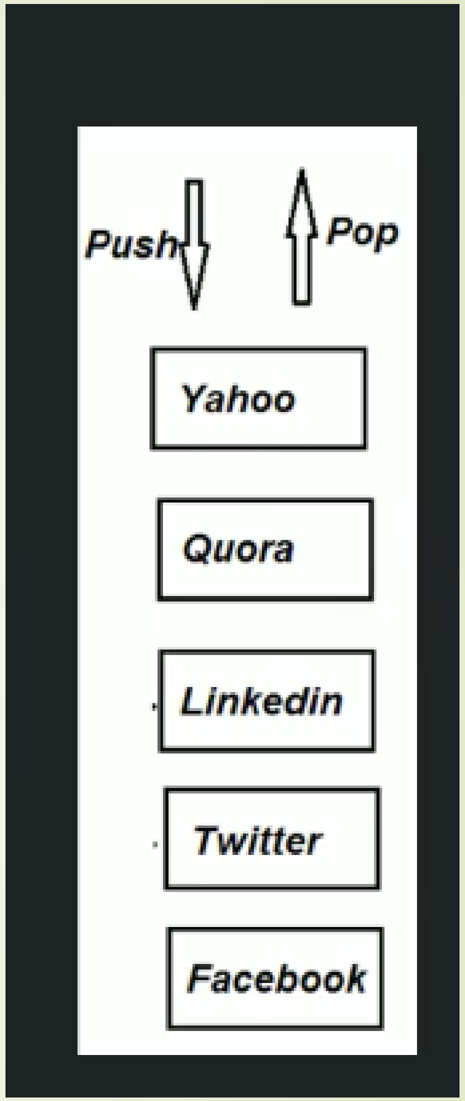
Why should we learn Stack ?

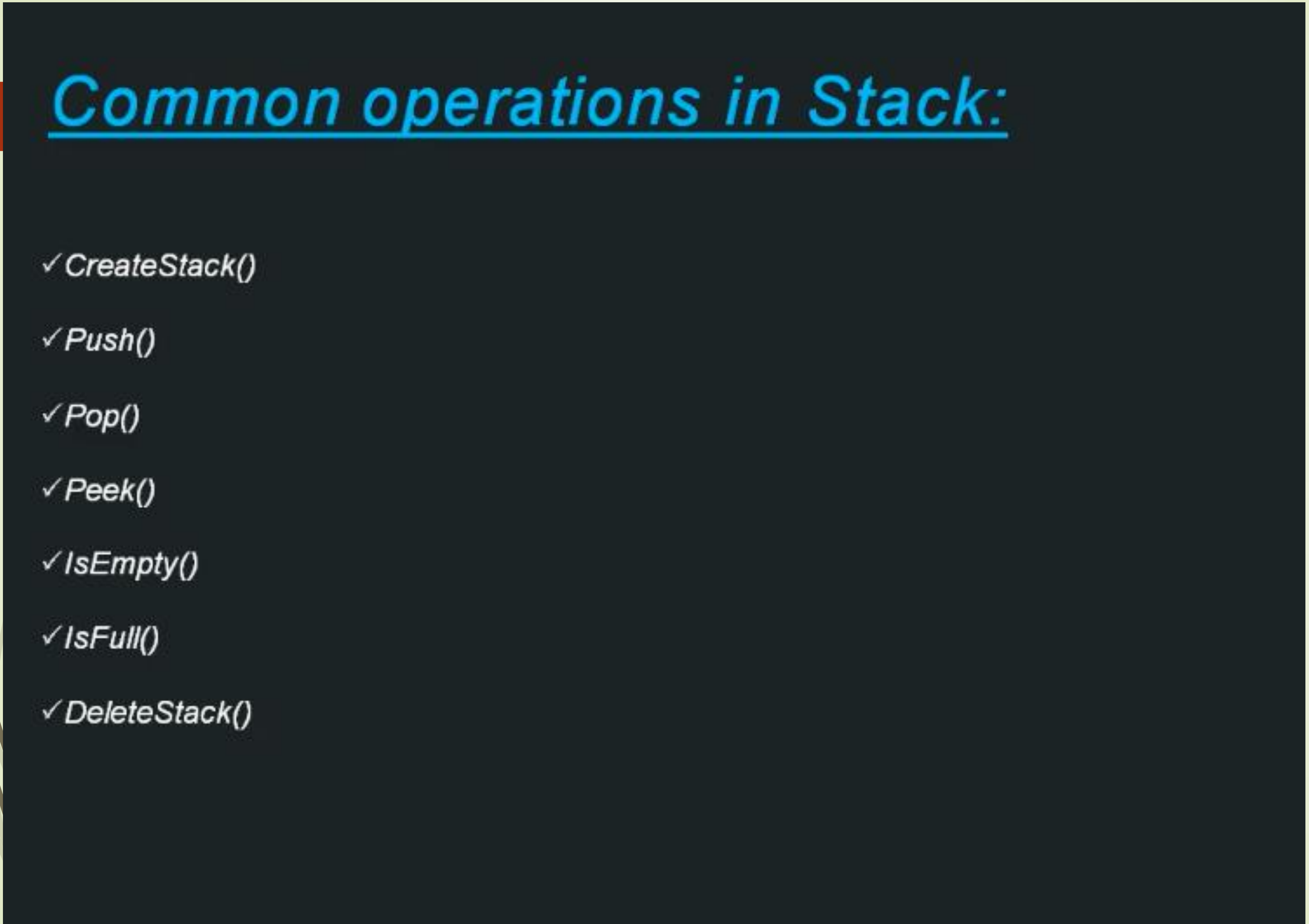
- ✓ *Why ?*
  - ✓ *When we need to create an application which utilizes 'last incoming data first'.*
  - ✓ *Example: implementation of 'back' button in browser.*

✓ *Example: implementation of 'back' button in browser.*




Facebook
Twitter
Linkedin
Quora
Yahoo

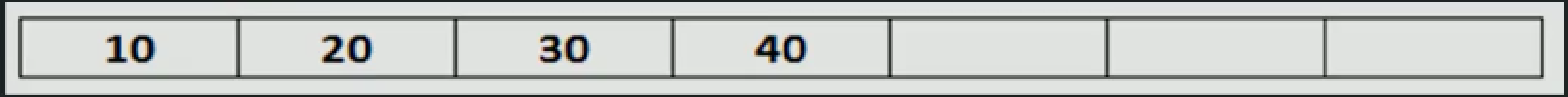

Push
Pop
Yahoo
Quora
Linkedin
Twitter
Facebook



Common operations in Stack:

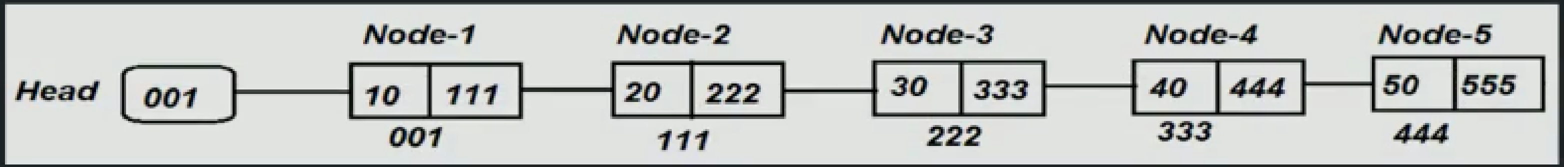
- ✓CreateStack()
- ✓Push()
- ✓Pop()
- ✓Peek()
- ✓IsEmpty()
- ✓IsFull()
- ✓DeleteStack()

Implementation options of Stack:

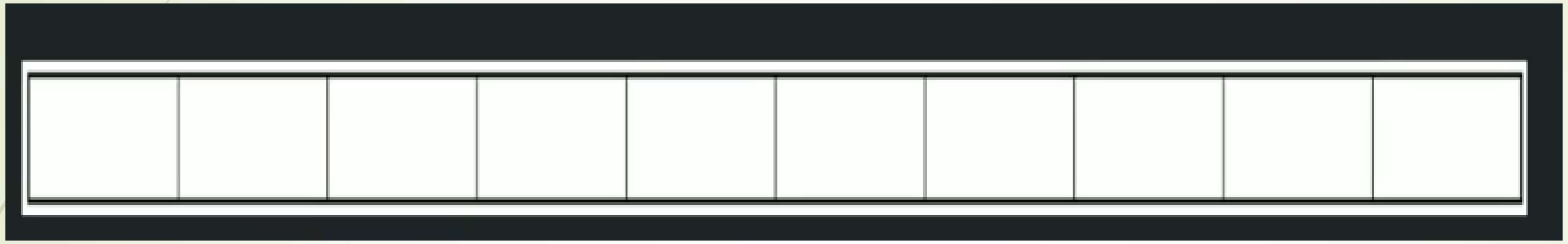


- ✓ Linked List:
  - ✓ Pros:
    - ✓ Variable Size
  - ✓ Cons:
    - ✓ Moderate in implementation

- ✓ Array:
  - ✓ Pros:
    - ✓ Easy to implement
  - ✓ Cons:
    - ✓ Fixed Size

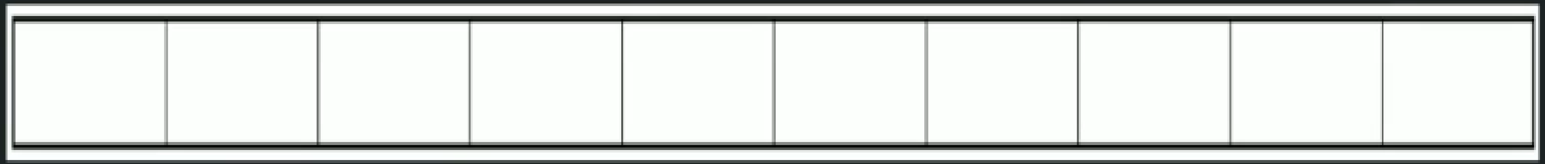


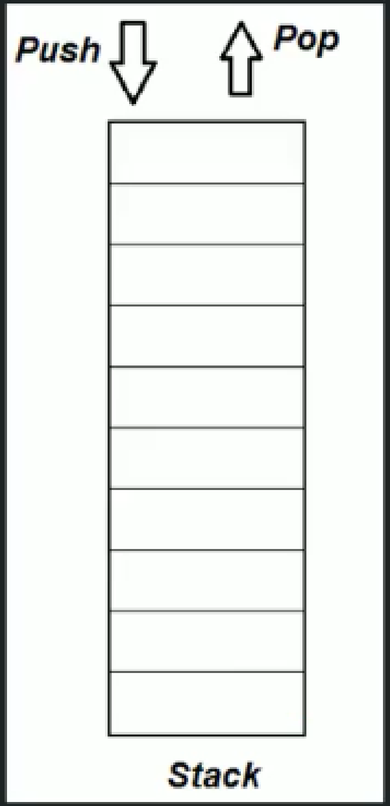
Implementation options of Stack:



```
CreateStack(int size):
    Create blank array of ‘size’
    Initialize variable “topOfStack” to -1
```

Push operation of Stack (Array implementation):

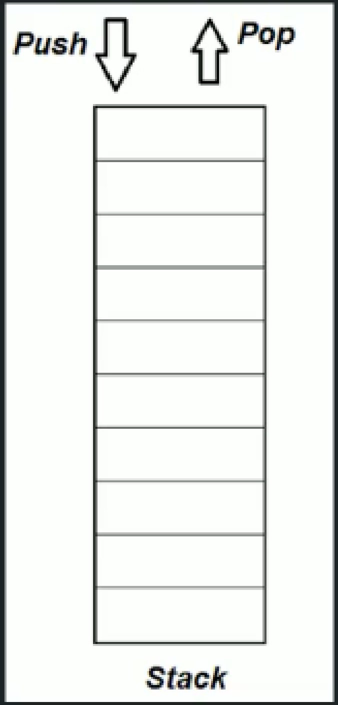




```
push (Value):
    if stack is full
        return error message
    else
        topofStack ++
        insert 'Value' at the top of the array
```

Pop operation of Stack (Array implementation):

```
pop()
    if stackisEmpty()
        return error message
    else
        print top of stack
        topOfStack--
```



Peek operation of Stack (Array implementation):

```
peek()
    if stackisEmpty()
        return error message
    else
        print topOfStack
```

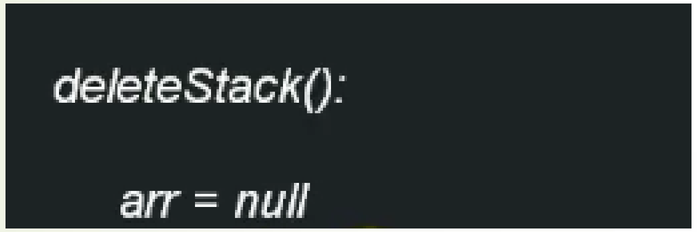
IsEmpty operation of Stack (Array implementation):

```
IsEmpty():
    if (topOfStack is -1)
        return true
    else
        return false
```

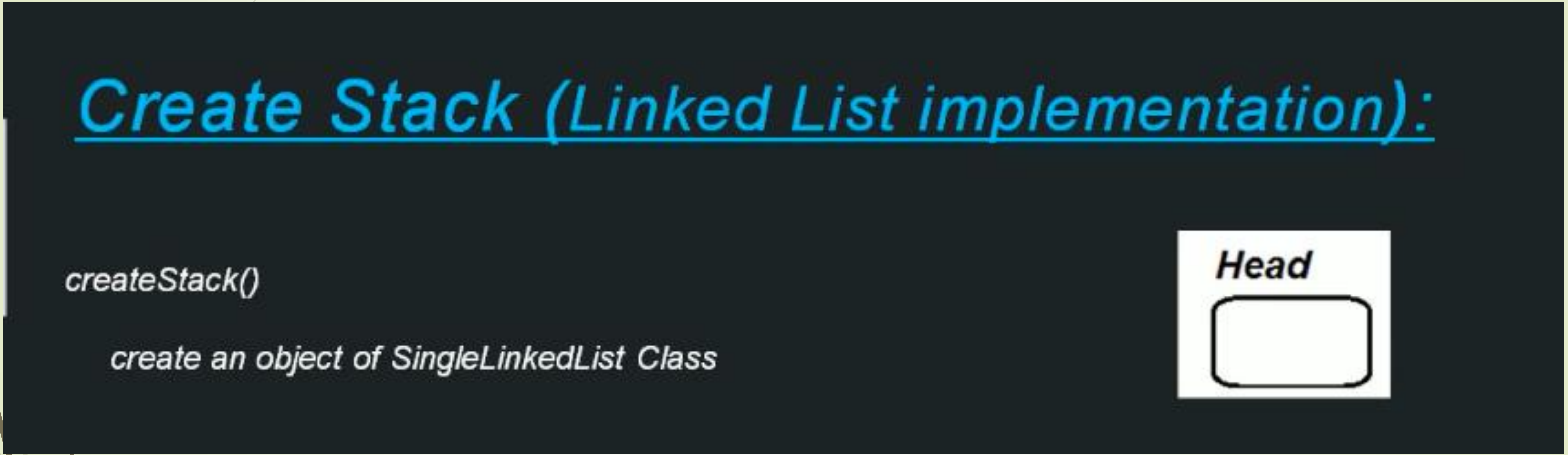
IsFull operation of Stack (Array implementation):

```
IsFull():
   if (topOfStack equals arr.size)
       return true
   else
        return false
```

Deletion of Stack (Array implementation):



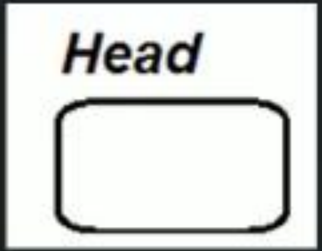
```
deleteStack():

    arr = null
```



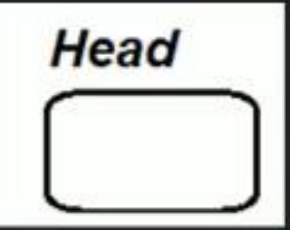
Create Stack (Linked List implementation):

createStack()

create an object of SingleLinkedList Class

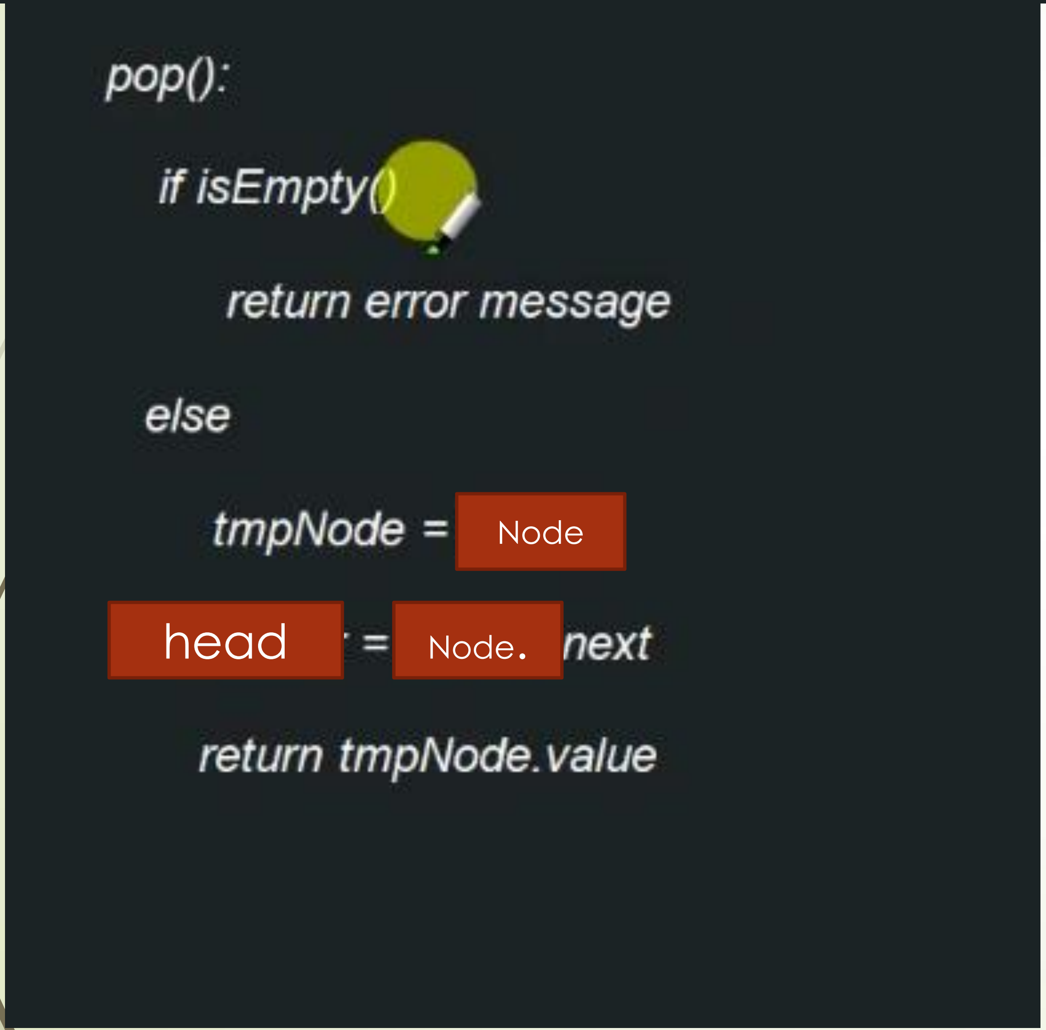


Push operation of Stack (Linked List implementation):



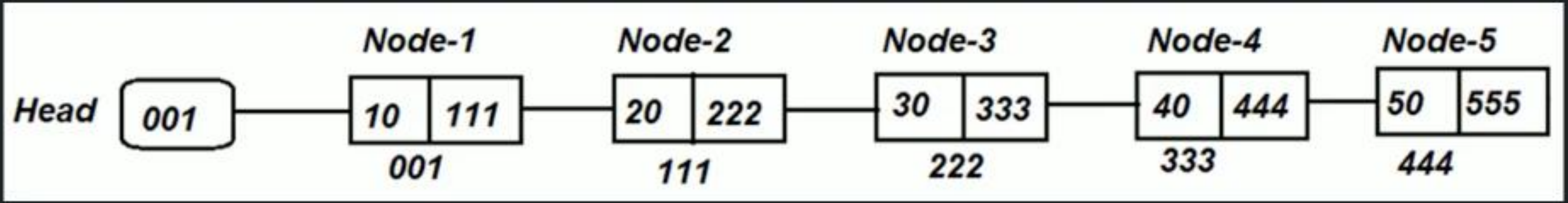
```
push(nodeValue):
    create a node
    node.value = nodeValue
    node.next = head
head = node
```

Pop operation of Stack (Linked List implementation):



```
pop():
  if isEmpty()
    return error message
  else
    tmpNode = Node
head = Node. next
    return tmpNode.value
```

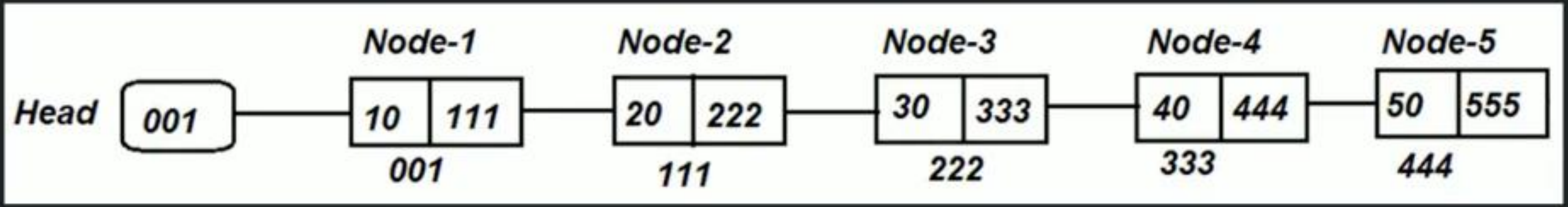
Peek operation of Stack (Linked List implementation



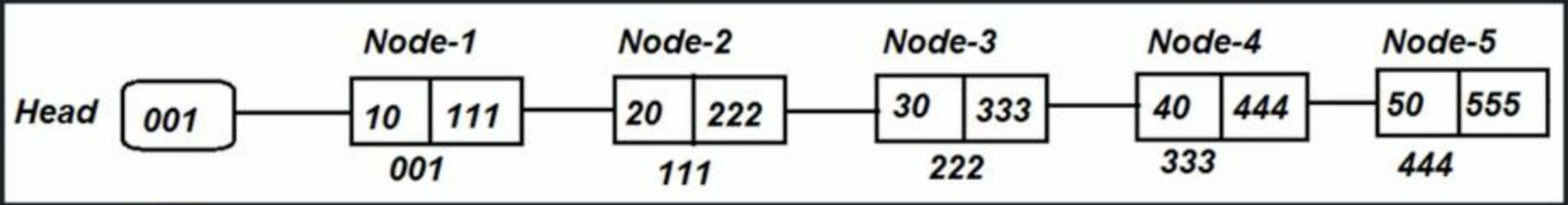
```
peek():
    return node.value
```

IsEmpty operation of Stack (Linked List implementation):

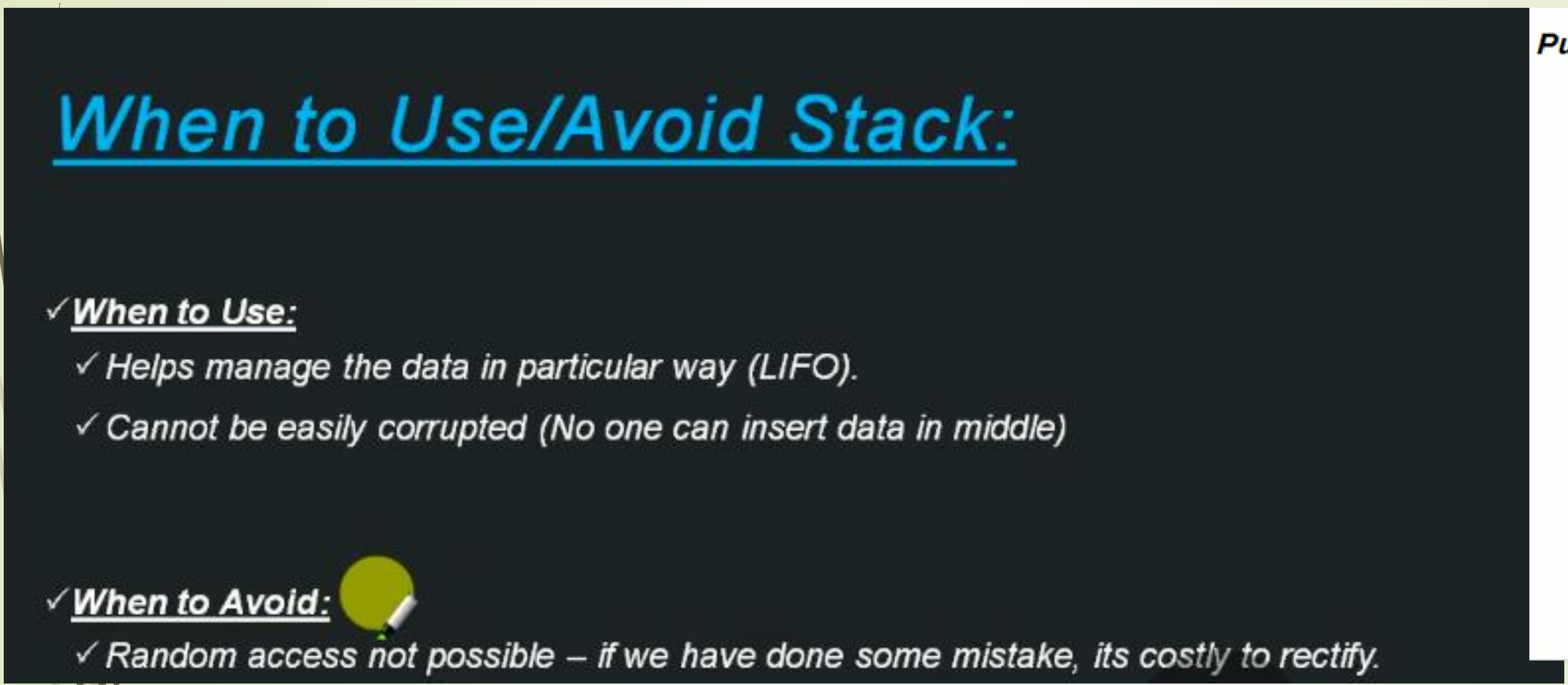
```
IsEmpty():
    if (header equals null)
        return true
    else
        return false
```



Deletion of entire Stack (Linked List implementation):

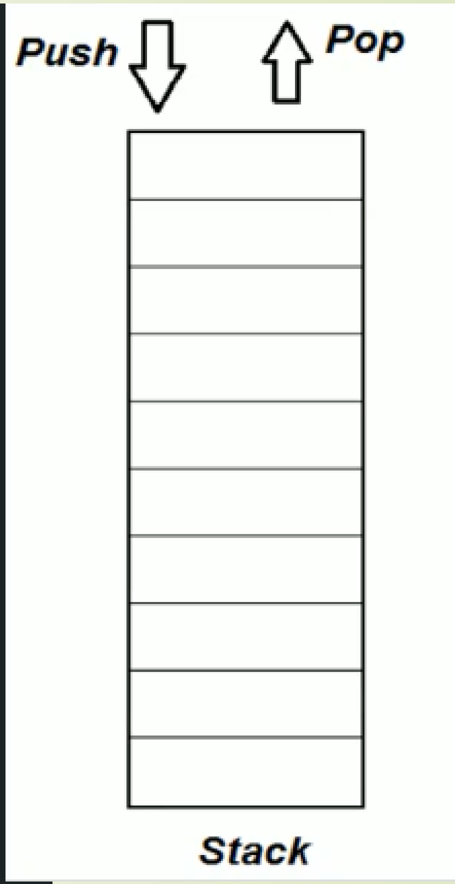


```
deleteStack():
  header = null
```



When to Use/Avoid Stack:

- ✓ **When to Use:**
  - ✓ Helps manage the data in particular way (LIFO).
  - ✓ Cannot be easily corrupted (No one can insert data in middle)
- ✓ **When to Avoid:**
  - ✓ Random access not possible – if we have done some mistake, its costly to rectify.



Thank you