UCS1302:
DATA STRUCTURES

Circular Linked list ADT

Session Meta Data

Author	**Dr. B. Bharathi**
Reviewer	
Version Number	1.2
Release Date	02 July 2019

Revision History

Revision Date	Details	Version no.
22 September 2017	1. New SSN template applied	1.2

Session Objectives

- To learn about Circular Linked list ADT
- Implementation of Circular Linked list

Session Outcomes

- At the end of this session, participants will be able to
  - Understand the concepts of Circular Linked list ADT
  - Implementation of Circular Linked list ADT

Agenda

- Circular Linked list ADT
- Implementation of Circular linked list operations

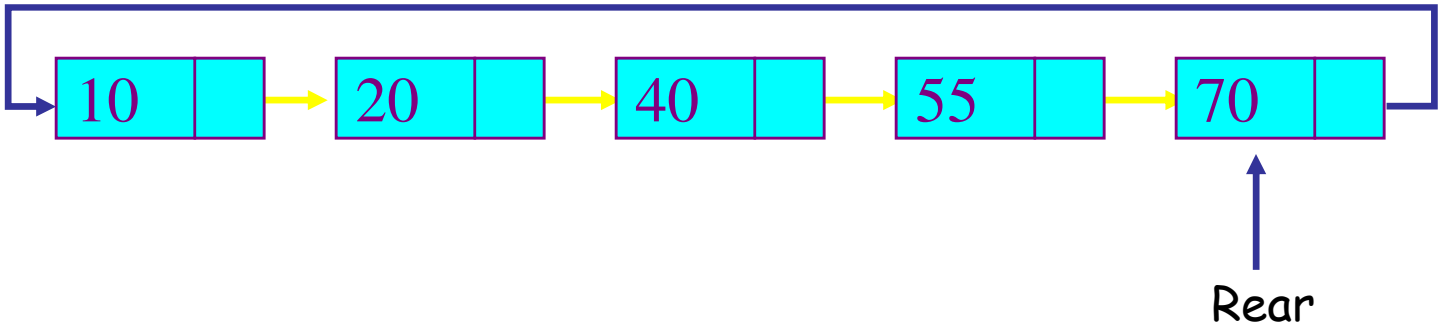
Circular Linked List ADT

Dr. B. Bharathi
SSNCE

July 02, 2019

Circular Linked Lists

- A Circular Linked List is a special type of Linked List
- It supports traversing from the end of the list to the beginning by making the last node point back to the head of the list
- A Rear pointer is often used instead of a Head pointer



Circular Linked List Definition

```
struct Node{
    int data;
    Node* next;
};
typedef Node* NodePtr;
```


Circular Linked List Operations

- insertNode(NodePtr& Rear, int item)
  - //add new node to ordered circular linked list

- deleteNode(NodePtr& Rear, int item)
  - //remove a node from circular linked list

- print(NodePtr Rear)
  - //print the Circular Linked List once

Traverse the list

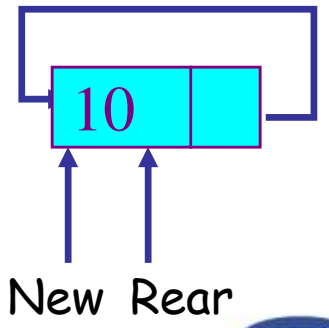
```
void print(NodePtr Rear){
    NodePtr Cur;
    if(Rear != NULL){
        Cur = Rear->next;
        do{
            printf("%d", Cur->data);
            Cur = Cur->next;
        }while(Cur != Rear->next);
    }
}
```

Insert Node

- Insert into an empty list

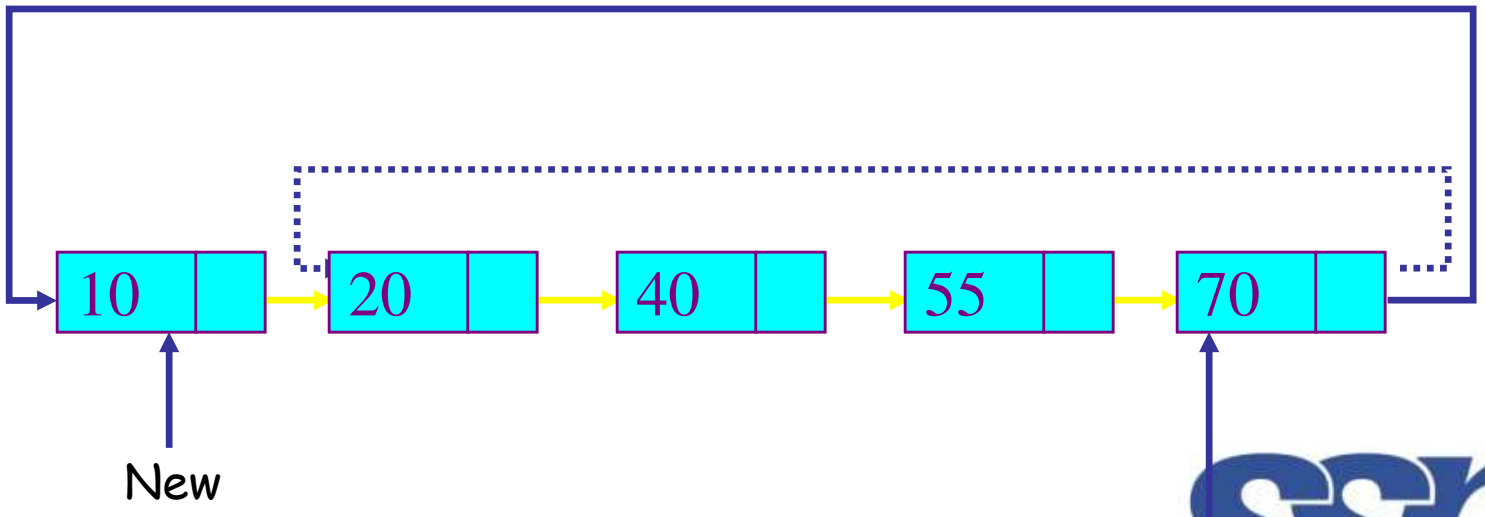
```
New = (Node*)malloc(sizeof(Node));
New->data = 10;

Rear = New;
Rear->next = Rear;
```



- Insert to head of a Circular Linked List

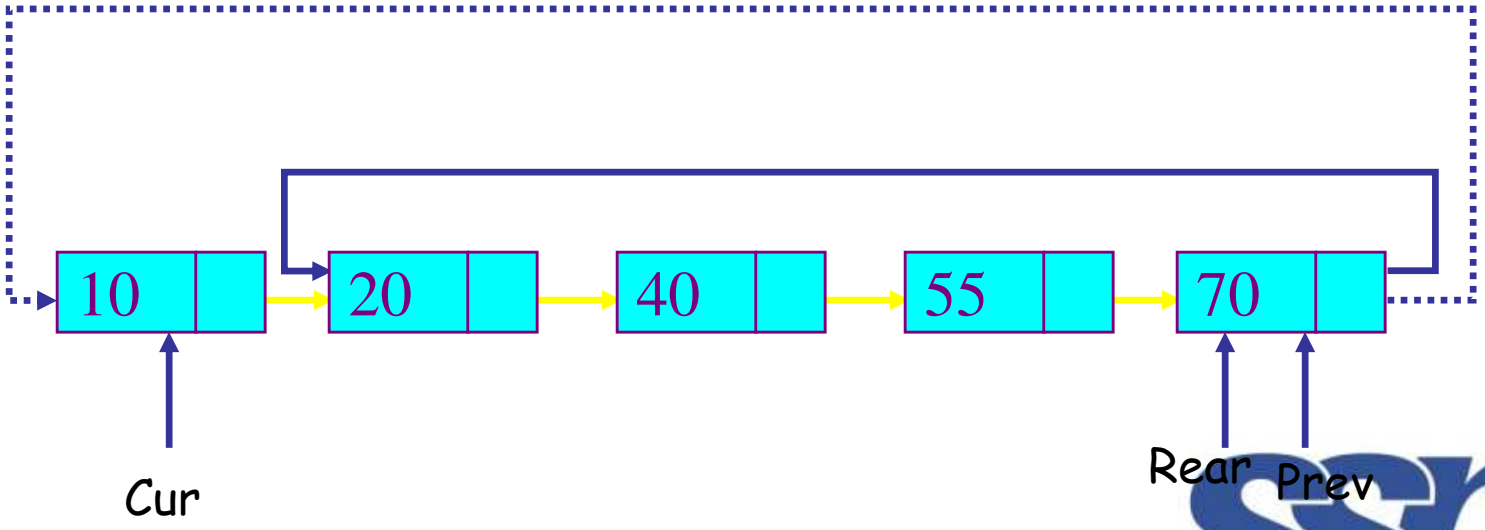
```
New->next = Rear->next;
Rear->next = New;
```



Delete Node

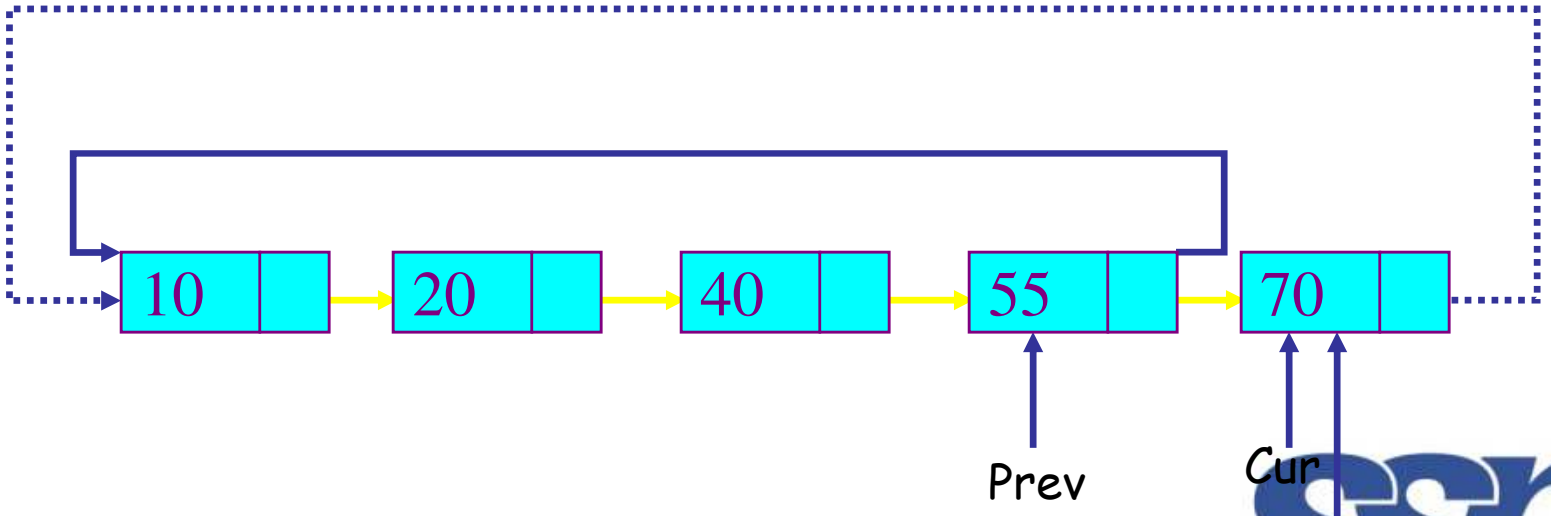
- Delete the head node from a Circular Linked List

```
Prev->next = Cur->next;     // same as: Rear->next = Cur->next
delete Cur;
```



- Delete the end node from a Circular Linked List

```
Prev->next = Cur->next;    // same as: Rear->next;
delete Cur;
Rear = Prev;
```



Summary

- Circular Linked list ADT
- Circular Linked list operations