

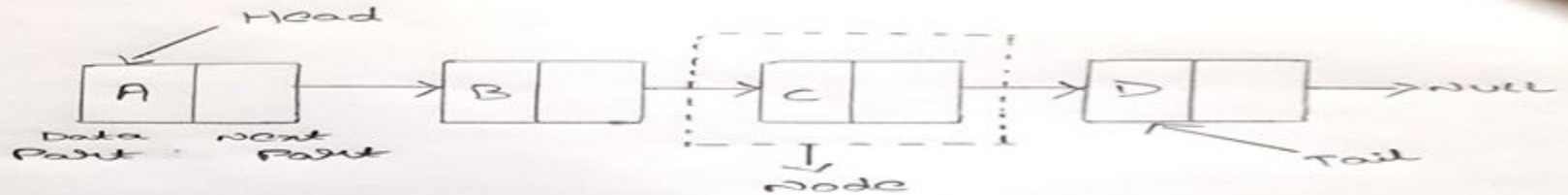
Linked List in Data Structure

K. Nanda Kishore

MS(IIITH)

Defining Linked List

- Linked List is an ordered collection of elements called nodes which has two parts.
- The Data part and Next part.
- The data contains elements.
- Next contains address of another node.



Data Part + Next Part = node.

Data Part = elements.

Next Part = Address of other node.

Linked List

Array vs. Linked List

Differences:

Array	Linked List
It is of fixed size.	It is dynamic and flexible.
Memory is assigned during compile time.	Allocated during runtime.
Operations like insertion and deletion consume a lot of time.	Consumes less time compared to that of an array.
Elements are stored consecutively in arrays.	Stored randomly.

Requirements & Terminologies

Requirements :

- Ability to grow - Add or Insert elements.
- Ability to shrink - Remove or Delete elements.
- Accessing list item - Access (or read) any element and to modify items (contents of node).
- Initialize - Clear or re-initialize list and to create an empty list.

Terminologies :

- Empty List.
- List Size.
- Head and Tail.
- Node.

Basic Operations

The basic operations on linked lists are:

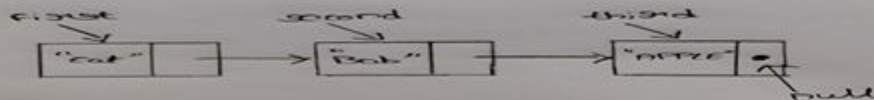
- Creation - The creation operation is used to create a linked list.

creating a linked list

```
Node third = new Node();  
third.item = "apple";  
third.next = null;
```

```
Node second = new Node();  
second.item = "Bob";  
second.next = third;
```

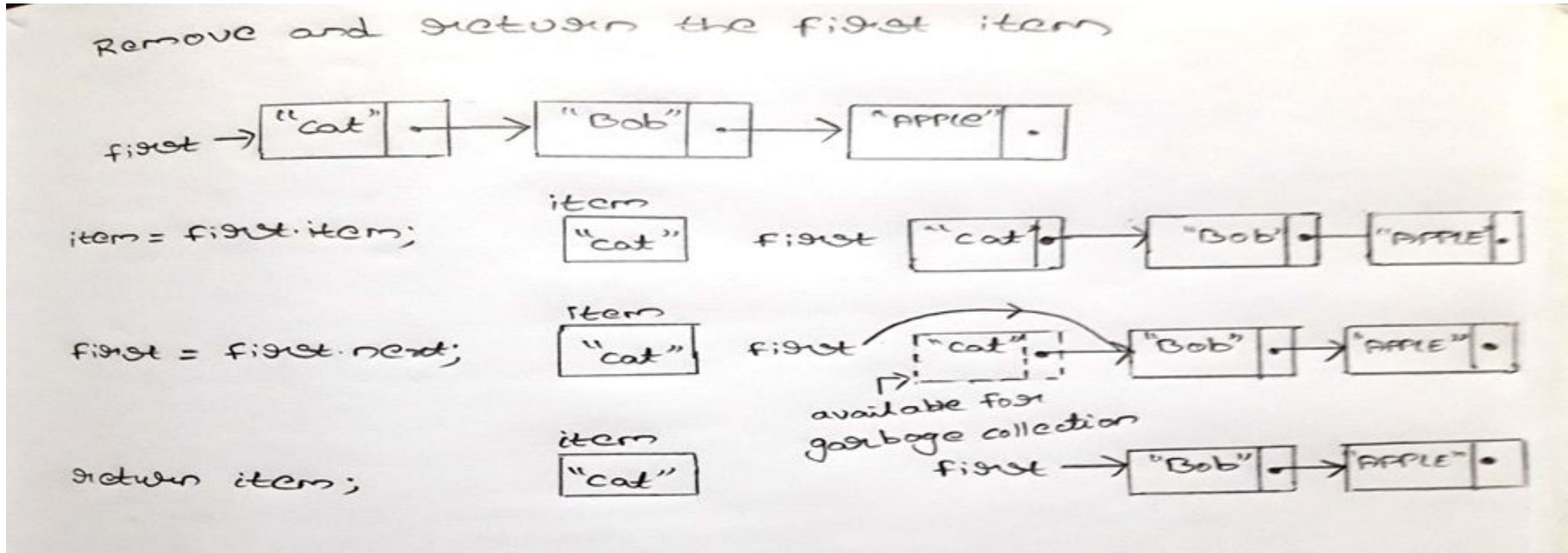
```
Node first = new Node();  
first.item = "cat";  
first.next = second;
```



addr	value
C0	"APPLE"
C1	null
C2	
C3	
C4	"cat"
C5	CA
C6	
C7	
C8	"Bob"
C9	C0

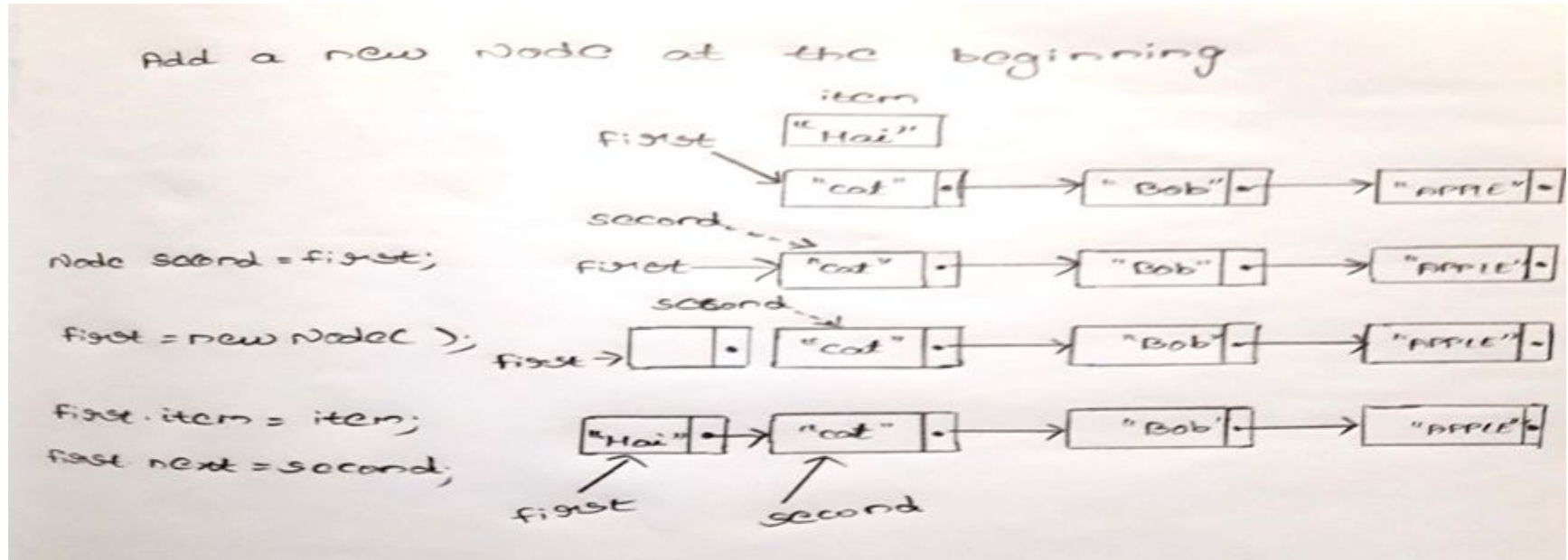
Basic Operations

- Remove - The remove operation is used to remove an item from the linked list.



Basic Operations

- Insertion - The insertion operation is used to insert a new node in the linked list at the specified position.



Basic Operations

- Traversing - The traversing operation is a process of going through all the nodes of a linked list from one end to another end.

Traversing

```
node x = first;  
while (x != null)  
{  
    System.out.println(x.item);  
    x = x.next;  
}
```

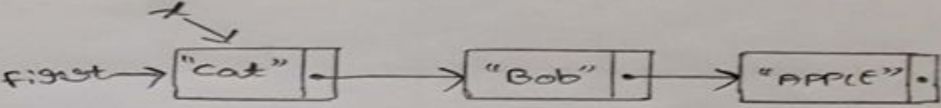


Diagram illustrating a linked list structure with three nodes: "Cat", "Bob", and "APPLE". A pointer labeled "first" points to the first node. A variable "x" also points to the first node. The nodes are connected sequentially by arrows.

Output:

```
cat  
Bob  
APPLE
```


Advantages of Linked Lists

- We can dynamically allocate memory space as needed.
- We can release the unused space in the situation where the allocated space seems to be more.
- Operation related to data elements like insertions or deletion are more simplified.
- Operation like insertion or deletion are less time consuming.
- Linked lists provide flexibility in allowing the items to be arranged efficiently.

Applications

In Computer Science:

- Implementation of stacks and queues.
- Dynamic memory allocation : We use linked list of free blocks.
- Maintaining directory of names.

In Real World:

- Image viewer – Previous and next images are linked, hence can be accessed by next and previous button.
- Previous and next page in web browser – We can access previous and next URL searched in web browser by pressing back and next button since, they are linked as linked list.
- Music Player – Songs in music player are linked to previous and next song. you can play songs either from starting or ending of the list.

Thank you

K. Nanda Kishore

MS(IIIth)