

DATE : \_\_\_ / \_\_\_ / \_\_\_

# ASSIGNMENT # 02

Name:- Manahil Afzam

Sap ID:- 68662

Course:- Data Structure

Date:- 21-Feb-2026

Instructor:- Nimra Waqas

Semester:- 3<sup>rd</sup>

RIPHAH INTERNATIONAL UNIVERSITY

GULBERG GREENS

## Question # 01

Explain the following elements of singly link list:

- Node
- Datafield
- Next pointer
- Head pointer.

Use diagrams to support your explanations.

$\Rightarrow$  Node :-

A node is a basic block of link list. It has two parts:

- Data (the actual value).
- Next pointer (address of next node).

Data	Next
------	------

$\Rightarrow$  Datafield :-

This part of the node stores the actual information like numbers, character and objects.

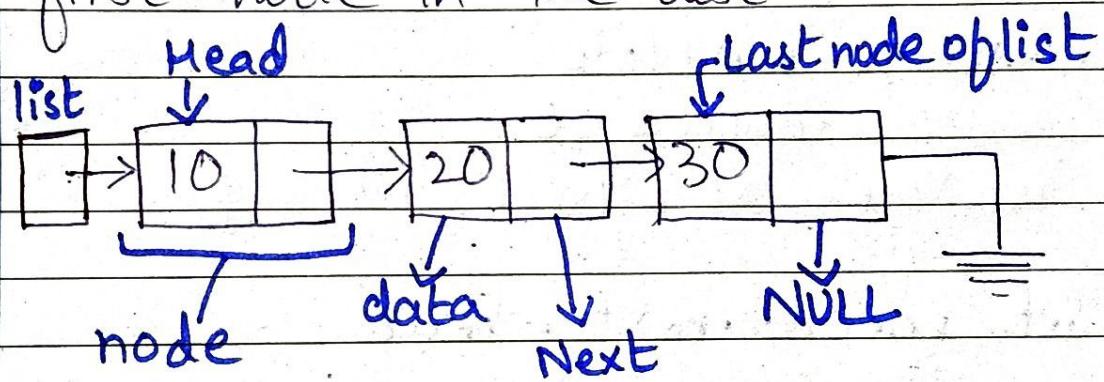
Example :- 10, 20, 30 etc.

⇒ Next Pointer:-

This is a reference (pointer) points to the next node in the field or list. If it's the last node, then, next points to null.

⇒ Head Pointer:-

The Head is a pointer that stores the address of the first node in the list.



## Part A

— Insertion Operations.

⇒ Write pseudocode and dry run for:-

1. Insertion at the Beginning.

Example: Insert 10 into list: 10, 30, 40.

Pseudocode:-

Procedure InsertAtBeginning (head, value):  
 newnode = Create Node (value).  
 newnode.next = head  
 head = newnode  
 return head  
 End Procedure

Dry Run:-

## 1- Create a new Node

- newnode.data = 10
- newnode.next = NULL (initially)

## 2- Set newnode.next = head

- newnode.next points to the current head (20).

## 3- Update head to newnode

- Head points to the newnode (10).
- Final list: head → 10 → 20 → 40 → NULL.  
 20 →

## 2- Insertion At the End:-

Example:- Insert 50 into list 10,20,30

Pseudocode:-

Procedure InsertAtEnd (head, value):  
 ptr = head

DATE: / /

```
while (ptr != NULL)
    ptr = ptr.next
```

```
p = CreateNode (value)
p.data = value
p.next = NULL
```

```
ptr.next = p
return head
```

End Procedure.

### Dry Run:-

- 1- **Move to next node:** Start from head(10)  
ptr = ptr.next, it moves towards  
the next node until you reach the  
last node (30).

- 2- **Create a newNode:**

- newNode.data = 50
- newNode.next = NULL.

- 3- **Link newNode at the end:-**

ptr.next = newNode → node 30 now  
points to 50.

Final list: 10 → 20 → 30 → 50 → NULL.

- 3- **Insertion at a Specific Position.**

Example: Insert 25<sup>at position 3</sup>  
into list: 10, 20, 30, 40

Pseudocode:-

Procedure InsertAtSpecificPosition(head, value):

    ptr = head  
 while ( $\overline{\text{ptr}} \overset{\text{data}}{\rightarrow} \neq x$ )  
      ptr = ptr.next

    p = newNode = CreateNode(value)

    newNode.data = value

    newNode.next = NULL

    p->next = ~~ptr~~.next

    ptr->next = p

End Procedure

Dry Run:-

1- Initialize Pointer

ptr = head

2- Search for node with data x.

$\text{ptr} \rightarrow \text{data} = 20$

loop stop and ptr now points to 20

3- Create newNode

p = newNode

$p \rightarrow \text{data} = 25$

$p \rightarrow \text{next} = \text{NULL}$

4- Link newNode into the list:

$\text{ptr} \rightarrow \text{next} = \text{ptr} \rightarrow \text{next}$  points to 30

DATE: \_\_\_/\_\_\_/\_\_\_

$\text{ptr} \rightarrow \text{next} = p$  node 20 now points to 25

Final list: 10 → 20 → 25 → 30 → 40 → NULL

## Part B: Deletion Operation.

### 4- Deletion from Beginning:-

Example: Delete first node from list: 10 → 20 → 30.

#### Pseudocode:-

```
Node *ptr  
ptr = head  
head = head->next  
delete ptr;
```

#### Dry Run:-

- Head points to node 10.
- Create a ptr pointer ( $\text{ptr} = \text{head}$  points to 10)
- Move head forward means head now points to 20  
 $\text{head} = \text{head} \rightarrow \text{next}$ .
- Unlink original head, now node 10 is isolated.
- Remove node 10
- Final list: head → 20 → 30 → NULL

## 5- Deletion from End

Example: Delete last node from  
list:  $10 \rightarrow 20 \rightarrow 30 \rightarrow 40$

Pseudocode:-

```

Node *ptr
Node *preptr
ptr = head
while (ptr->next != NULL)
    preptr = ptr
    ptr = ptr ->next
preptr->next = NULL
delete ptr.

```

Dry Run:-

- Create two pointers  $\text{ptr}$  and  $\text{preptr}$ .
- $\text{ptr} = \text{head}$  points to 10
- Move forward until you reach the last node 40.
- $\text{preptr} = \text{ptr}$ , now it points 30
- $\text{ptr}->\text{next}$  points to NULL, it shows that the last node is 30.
- At the end, delete  $\text{ptr}(40)$ .
- Final list:  $10 \rightarrow 20 \rightarrow 30 \rightarrow \text{NULL}$ .

## 6- Deletion from a specific position.

Example:- Delete node at position 2 in the list:  $10 \rightarrow 20 \rightarrow 30 \rightarrow 40$

Pseudocode:-

```

Node *ptr
Node *preptr
ptr = head
while (ptr->data != x)
    preptr = ptr
    ptr = ptr->next
    preptr->next
    preptr->next = ptr->next
    delete ptr

```

DryRun:-

- Create two pointers ptr and preptr
  - ptr = head points to 10
  - ptr moves forward until it reaches the node x (20).
  - preptr->next = ptr->next points to 30 (bypass 20)
  - Remove ptr means delete node 20.
  - Final list - 10 → 30 → 40 → NULL.
-