PES UNIVERSITY

DATA STRUCTURES AND ITS APPLICATIONS

Vandana M L

Department of Computer Science and Engineering

DATA STRUCTURES AND ITS APPLICATIONS

Singly Linked List

Vandana M L

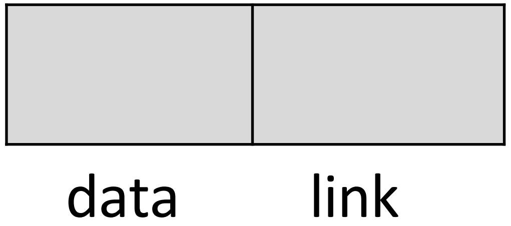
Department of Computer Science and Engineering

Singly Linked List: Node Structure

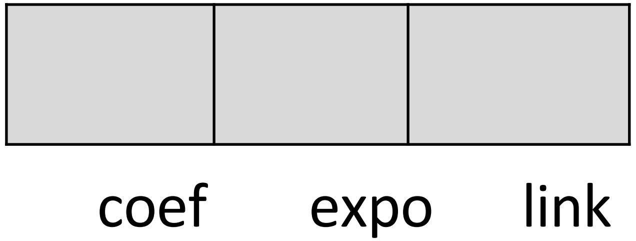
Defining node structure

```
struct node
{
int data;
struct node *link;
};

typedef struct node NODE;
```



```
struct polydata
{
  int coeff;
  int expo;
};
```

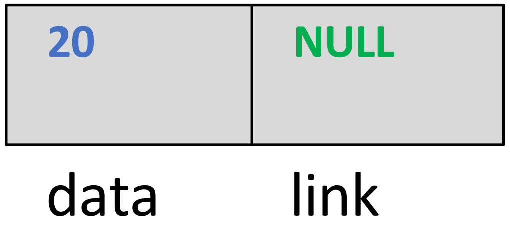


```
typedef struct node
 {
polydata data;
struct node *link;
}polynode;
```

Creating a node

- ➢ Allocate memory for the node dynamically

- ➢ If the memory is allocated successfully
  set the data part to user defined value
  set the link part to NULL



Inserting a node

There are 3 cases

- Insertion at the beginning
- Insertion at the end
- Insertion at a given position

Insertion at the beginning

- Create a node

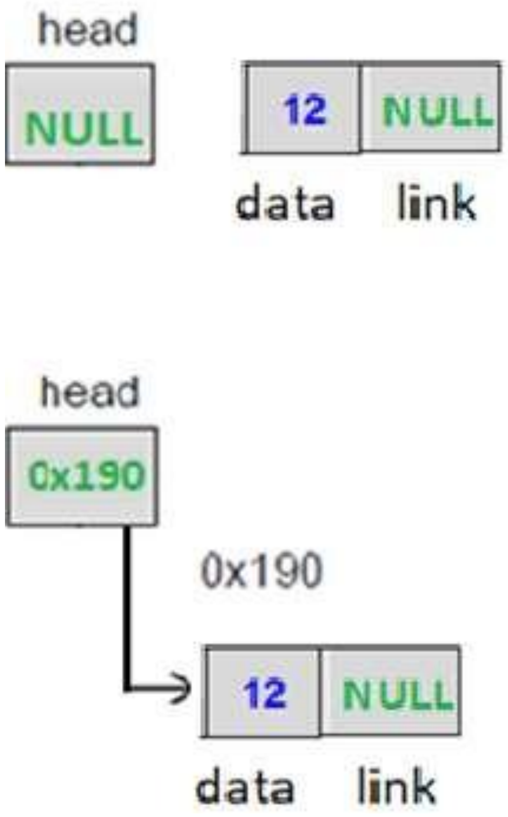
If the list is empty

- make the head pointer point towards the new node;

Else

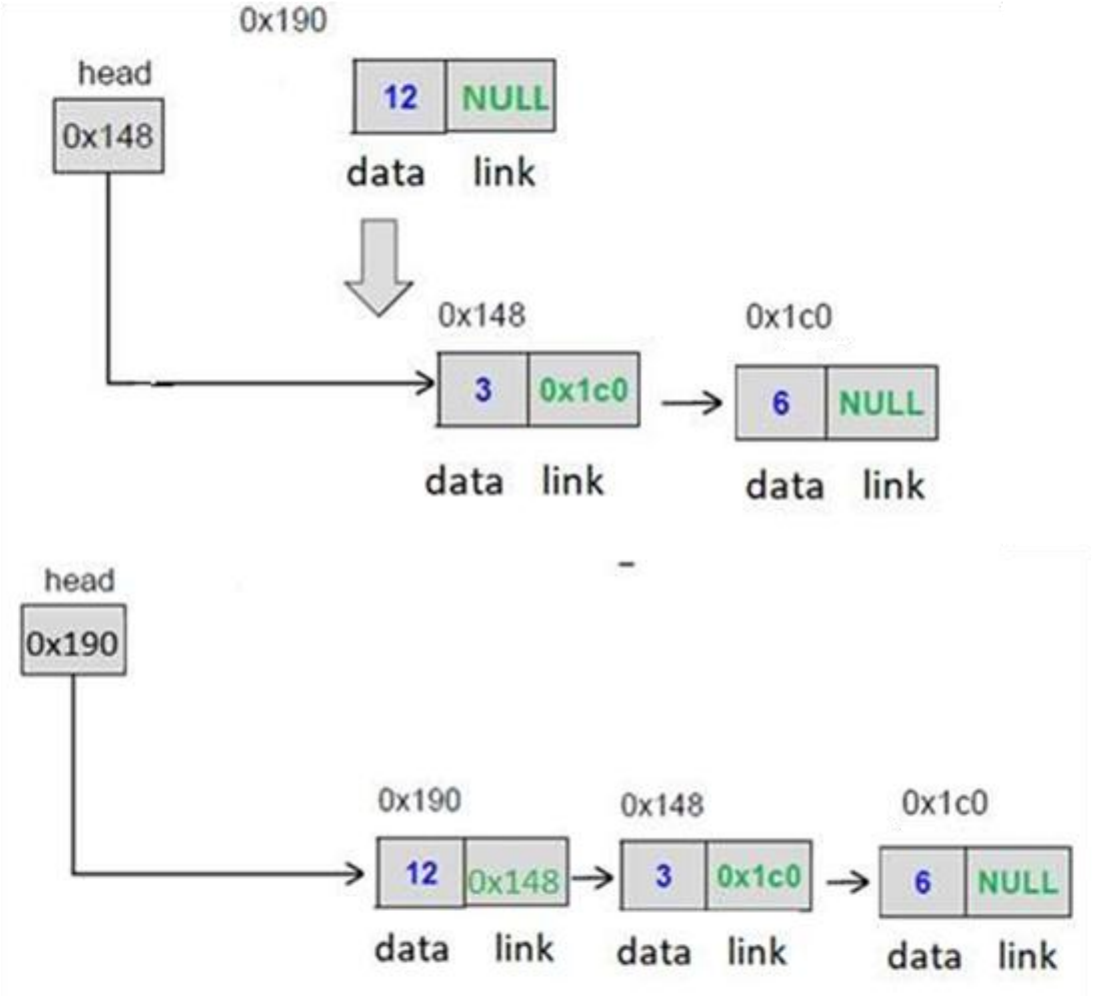
- Make the next pointer of the node point towards the first node of the list
- Make the head pointer point towards this new node

Insertion at the beginning



CASE1

Insertion at the beginning



CASE2

Insertion at the end of the

- Create a node

If the list is empty

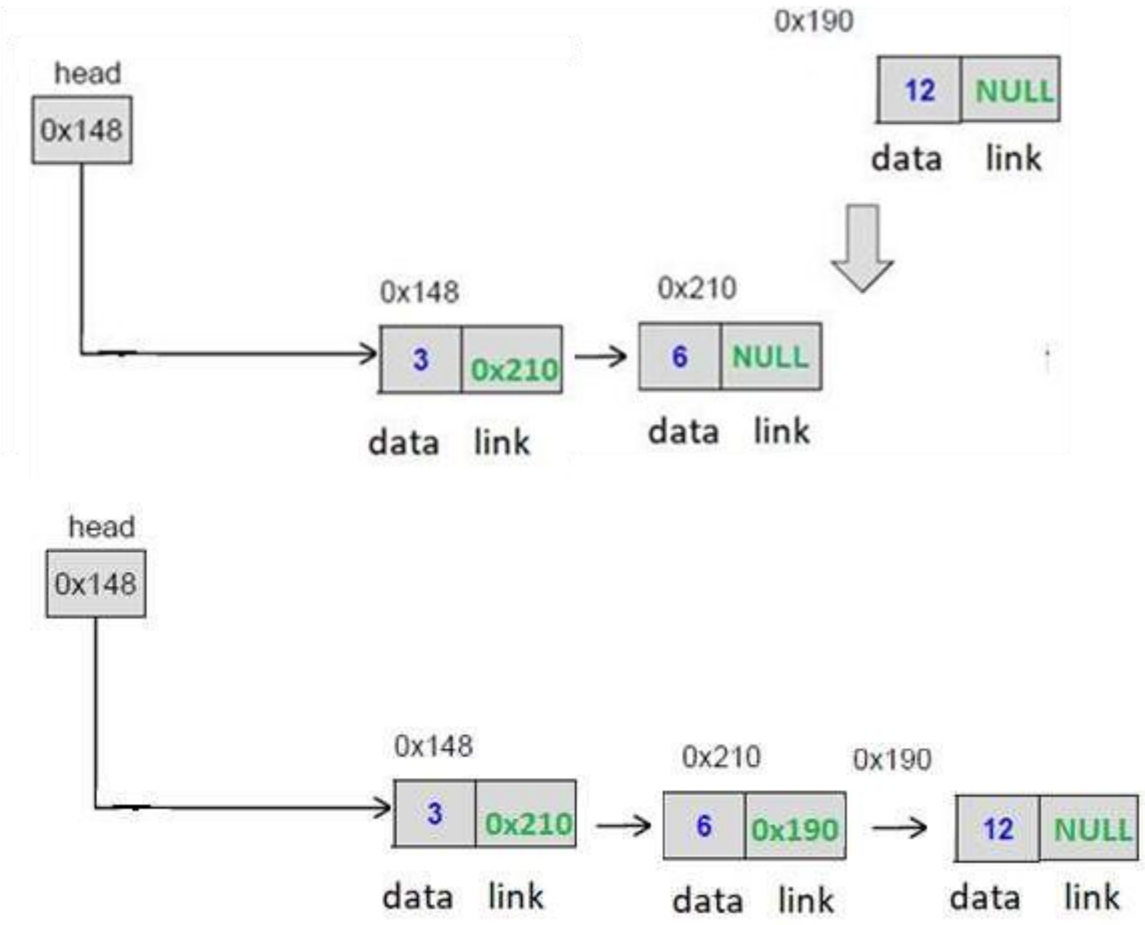
- make the head pointer point towards the new node;

Else

- Traverse the linked list to find out the last node
- Make the link pointer of the last node to point to the new node

Singly Linked List operations

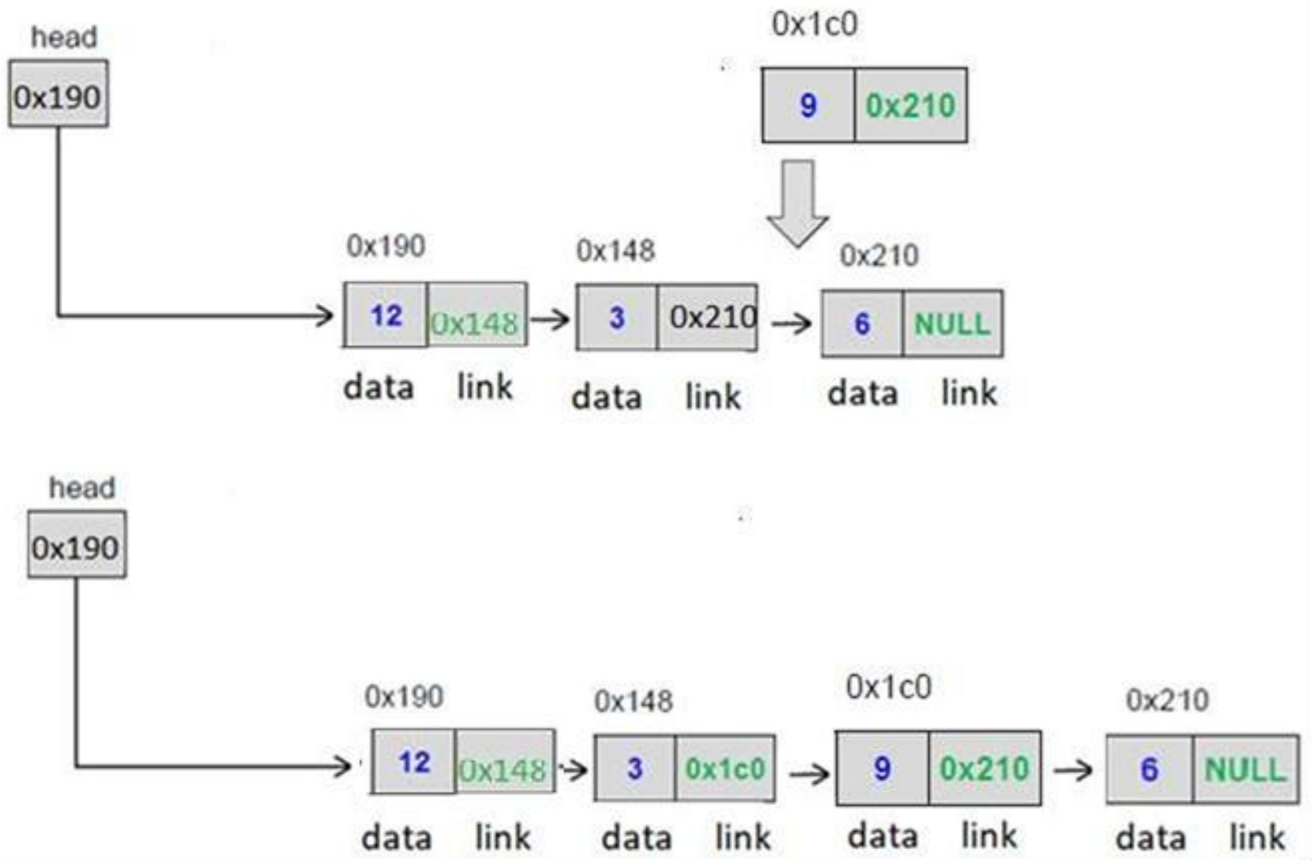
Insertion at the end of the



Insertion at the given position

```
- Create a node
If the list is empty or if insertion is to be done at first position
- Same steps as insert front
Else
- Traverse the linked list to reach given position
- Keep track of the previous node
    If it is an intermediate position
    - Change previous node link to point to the newnode
    - Newnode to point to the next node
    Else
        If it is last position
        - Same steps as insert at end
        Else
            Print “Invalid position”
```

Insertion at the given position



Singly Linked List insert operation

Apply the concepts to implement following operations on a circular singly linked list

- Count number of nodes
- Concatenate two lists
- Sum of all the node values in the list

1. What is the correct sequence of steps to create a new node and insert it at the beginning of an SLL?

a) Allocate memory → Set first node to new node→ Set data → Update head

b) Set data → Allocate memory → Set new node’s next to head → Update head

c) Allocate memory → Set data → Set new node’s next to head → Update head

d) Update head → Set data → Allocate memory

1. What is the correct sequence of steps to create a new node and insert it at the beginning of an SLL?

a) Allocate memory → Set first node to new node→ Set data → Update head

b) Set data → Allocate memory → Set new node's next to head → Update head

c) Allocate memory → Set data → Set new node's next to head → Update head

d) Update head → Set data → Allocate memory

2. When inserting a new node at the head of a singly linked list, what is the correct sequence of operations?

a) new->next = head; head = new;

b) head = new; new->next = head;

c) new->next = NULL; head = new;

d) new->next = head->next; head = new;

2. When inserting a new node at the head of a singly linked list, what is the correct sequence of operations?

a) new->next = head; head = new;

b) head = new; new->next = head;

c) new->next = NULL; head = new;

d) new->next = head->next; head = new;

3. To insert a new node after a given node p in an SLL, which step is incorrect?

a) Allocate memory for the new node.

b) new->next = p->next;

c) p->next = new;

d) new->next = p;

3. To insert a new node after a given node p in an SLL, which step is incorrect?

a) Allocate memory for the new node.

b) new->next = p->next;

c) p->next = new;

d) new->next = p;

4. What happens when you insert a node at the end of an empty SLL without checking if head == NULL?

a) Segmentation fault due to NULL->next access.

b) Node is automatically added as head.

c) Compiler error occurs.

d) It works if malloc is used.

4. What happens when you insert a node at the end of an empty SLL without checking if head == NULL?

a) Segmentation fault due to NULL->next access.

b) Node is automatically added as head.

c) Compiler error occurs.

d) It works if malloc is used.

5. Which of the following correctly inserts a node at position n (1-based indexing) in an SLL?

a) Traverse n-1 nodes and then insert.
b) Traverse n nodes and then insert.

c) Always insert at head if n > 1.

d) Insert by swapping data of nodes.

5. Which of the following correctly inserts a node at position n (1-based indexing) in an SLL?

a) Traverse n-1 nodes and then insert.

b) Traverse n nodes and then insert.

c) Always insert at head if n > 1.

d) Insert by swapping data of nodes.

6. In a singly linked list, inserting a new node before a given node p (not head) requires:

a) Directly linking new->next = p and updating prev->next = new, where prev is the previous node of p.

b) Simply new->next = p without finding prev.

c) Swapping data between p and the new node (without using prev).

d) It is not possible to insert before a given node in an SLL.

6. In a singly linked list, inserting a new node before a given node p (not head) requires:

a) Directly linking new->next = p and updating prev->next = new, where prev is the previous node of p.

b) Simply new->next = p without finding prev.

c) Swapping data between p and the new node (without using prev).

d) It is not possible to insert before a given node in an SLL.

7. If an SLL has n nodes, how many pointer changes are required to insert a new node after the k-th node (1 ≤ k ≤ n)?

a) 1

b) 2

c) k

d) n

7. If an SLL has n nodes, how many pointer changes are required to insert a new node after the k-th node (1 ≤ k ≤ n)?

a) 1

b) 2

c) k

d) n

THANK YOU

Vandana M L

Department of Computer Science & Engineering

vandanamd@pes.edu