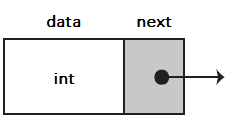
**Linked Lists (Single)**



struct Node

{

int data; //some data

Node \*next; //pointer to next Node

};

**Creating a Node**

//main

Node\* newNode = new Node;

int value;

cout << "Enter the data " ;

cin >> value;

newNode = createNode(value); //function call

//function

Node\* createNode(int value)

{

Node \*temp = new Node;

temp->data=value;

temp->next=NULL;

return temp;

}

**Printing the List**

void printList(Node \*temp)

{

while (temp != NULL)

{

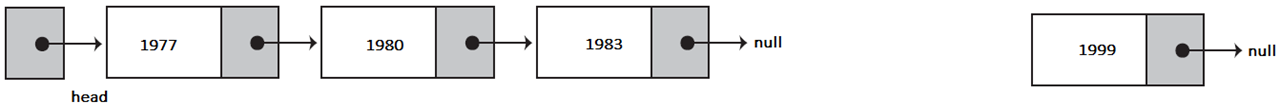
cout << temp->data << endl;

temp = temp->next; //move to next Node in the list

}

}

**Add a Node to the End**



//main

appendList(head, newNode);

//function to add to the end

void appendList(Node \*head, Node \*newNode)

{

newNode->next=NULL;//set newNode pntr to NULL

Node\* temp= new Node;

if (!head)

{

head=newNode;

}

else

{

temp=head;

while(temp->next)

{

temp=temp->next;//moves through list

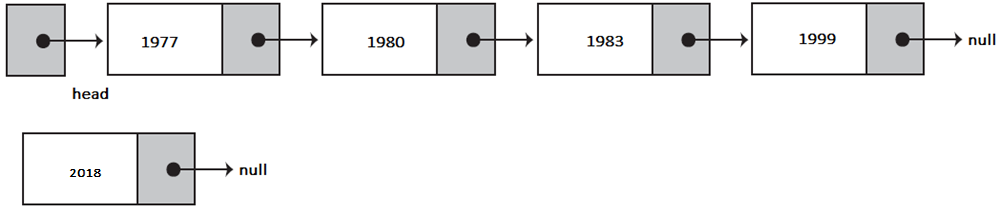
}

temp->next=newNode;

}

}

**Add a Node to the Beginning**

****

//main

head = appendFront(head, newNode); **//function call**

//function to add to the beginning

Node\* appendFront(Node\* head, Node\* newNode)

{

If (head==NULL){

Head=newNode;

}

else{

newNode->next=head;

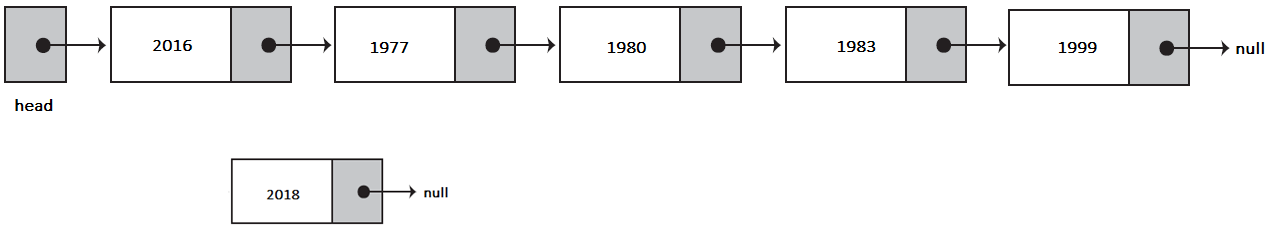
head=newNode;

}

return head;

}

**Inserting a Node at a Specific Position**



//insert at position 2

int pos = 2;

insertList(anotherNode, head, pos); **//function call**

//function to insert at a position

void insertList(Node \*randomnode, Node \*head, int position)

{

Node \*previous = new Node;

Node \*current=new Node;

current=head;

for(int i=0;i<position;i++)

{

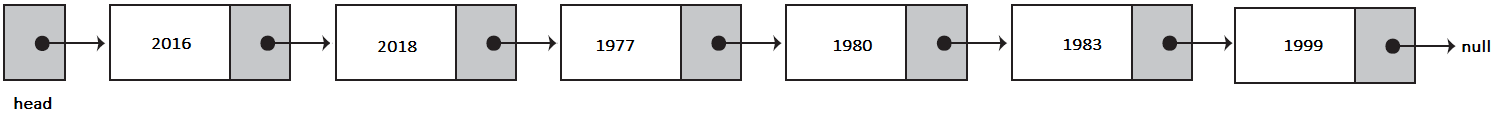
previous=current;

current=current->next;

}

}

**Delete First Node**



//main

head = deleteFirst(head); //function call

//function to delete head

Node\* deletefirst(Node \*head)

{

Node \*tmp=newNode

Tmp=head;

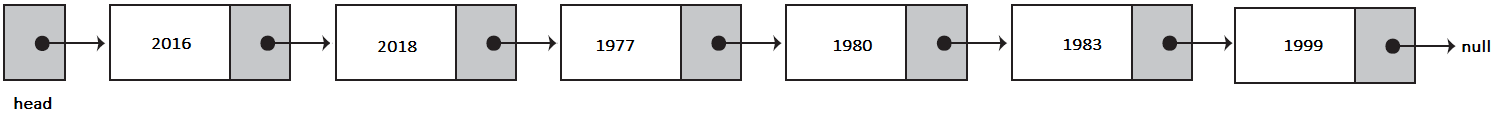
head=head->next;

delete tmp;

return head;

}

**Delete a Node at a Specific Position**



//main (delete position 2)

pos = 2;

deletePosition(head, 5); //function call

//function