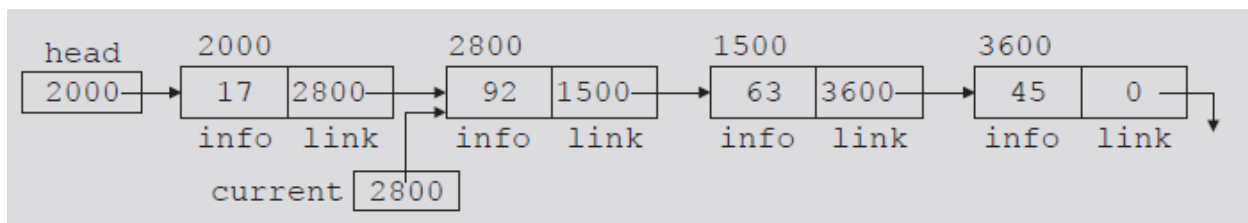


**The School Electrical Engineering and Information Technology
Computer Science Department**

**CS223 Lab
Linked List**

Definition:

- Linked list is a collection of components (nodes) such that every node contains data and address of the next node. Eg, The following is a linked list with 4 nodes:



A structure, which contains a data element and a pointer to the next node, is created as follows:

```
struct Node {  
    int value;  
    struct list *next;  
};
```

Main Operations

- Insertion

Adding a new node to the linked list in the beginning, end, or in between two nodes and adjusting the necessary connections.

- Deletion

Removing a given node from the linked list and adjusting the necessary connections.

- Traversal

Visiting each node once for doing something to that node, such as displaying the data in the node.

- Searching

Finding the location of a given item of information in the linked list and returning a pointer to the node.

Lab Work

Consider the following C++ code which contains the following operations:

- Define a structure node
- Declare Head and Tail nodes
- Create the linked list by adding a node to the beginning of a list
- Traverse the linked list to printing it

```
#include <iostream>
using namespace std;
struct Node
{
    int data;
    Node *next; // Pointer to next node
};

struct Node *head=NULL; //Declare Head Pointer
struct Node *tail=NULL; //Declare Tail pointer

// Insertion
void insertbeginning(int n){
    struct Node *newNode=new Node;
    newNode->data=n;
    newNode->next=head;
    head=newNode;
}

/*
void insert_last(int n){
    .
    .
    .
}
//delete node
void delete_at_position(){
    .
    .
    .
}

*/

//Traverse (print nodes data)
void traverse(){
    if(head==NULL){
        cout<<"Empty List!"<<endl;
        return;
    }
}
```

```

    }
    struct Node *temp=head;
    while(temp!=NULL){
        cout<<temp->data<<" ";
        temp=temp->next;
    } }

int main()
{
    insertbeginning (4);
    insertbeginning (2);
    traverse();
    return 0;
}

```

Lab Exercises:

1. Compile and run the previous code.
2. Fill in the blank the following functions :
 - a. Insert function **to insert a node at the end** of the linked list.
 - b. Delete function **to delete a node at any position** of the list.
3. After you finish please submit the final code to Elearning.