Riphah International University
I-14 Main Campus
Faculty of Computing

Class:	Fall-2024	**Subject:**	Data Structures & Algorithms
Course Code:	CS 2124	**Lab Instructor:**	Zeeshan Ali

Learning Objective:

- Link list
- Why link list?
- Node Class in C++
- Linkedlist Class in C++
- Operations on Linklist
- insertNode
- printList
- deleteNode
- Advantages
- Disadvantages
- Lab Task

Link list

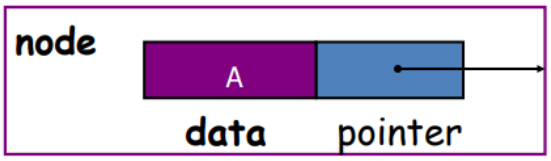
Collection of same type of objects.

It can be:

- Linked list
- Double linked list
- Circular linked list

Linked list is a sequence of links which contains items. Each link contains a connection to another link.

- A **link list** is a series of **connected nodes.**
- A data item **plus pointer** is called a **node.**
- **Head:** pointer to the first node.
- **Last node**: points to **NULL.**



Why link list?

Problem with array fixed length.
To expand an Array
Create a new array, longer in size and copy the contents of the old array into the new array
Deletion

Solution

Linked list

A linked list, or one-way list, is a linear collection of data elements, called nodes, where the linear order is given by means of pointer.

Node Class in C++

• We use two classes: **Node** and **List**

Declare Node class for the nodes

– data: int-type data in this example

– next: a pointer to the next node in the list

```
1  class Node{
2  public:
3      int data;
4      Node* next;
5  };
```

Linkedlist Class in C++

```
10  class Linkedlist {
11      Node* head;
12  public:
13      // Insert a node at the end of the linked list.
14      void insertNode(int);
15      // Print the linked list.
16      void printList();
17      // Function to delete the node at given position
18      void deleteNode(int);
19  };
```

Operations on Linklist

– **insertNode** : Insert a node at the end of the linked list.

–**printlist** : Print all nodes in the linked list.

–**deleteNode**: Delete the node at given position.

insertNode

```
35  void Linkedlist::insertNode(int data)
36  {
37      // Create the new Node.
38      Node* newNode = new Node(data);
39      // Assign to head
40      if (head == NULL) {
41          head = newNode;
42          return;
43      }
44      // Traverse till end of list
45      Node* temp = head;
46      while (temp->next != NULL) {
47          // Update temp
48          temp = temp->next;
49      }
50      // Insert at the last.
51      temp->next = newNode;
52  }
```

printList

```
53  // Print all the nodes of the linked list.
54  void Linkedlist::printList()
55  {
56      Node* temp = head;
57      // Check for empty list.
58      if (head == NULL) {
59          cout << "List is empty" << endl;
60          return;
61      }
62      // Traverse the list.
63      while (temp != NULL) {
64          cout << temp->data << endl;
65          temp = temp->next;
66      }
67  }
```

deleteNode

Advantages:

- Stack and queue can be implemented
- Insertion and deletion is fast
- Addition and removal from the middle
- No need to define initial size of list

Disadvantages

- More memory than arrays
- Sequential order, read data from beginning.
- Takes time to access individual element.

Lab Task.

Write a program to get input [NAME and SAP_ID of student] from user insert input in linked-list. Delete record of 2nd and 5th student.
Note: Number of inputs at least 5.