National University of Computer & Emerging Sciences (NUCES) Islamabad,
Department of Computer Science

DATA STRUCTURES – FALL 2021

LAB 05

Singly Linked List

Learning Outcomes

In this lab you are expected to learn the following:

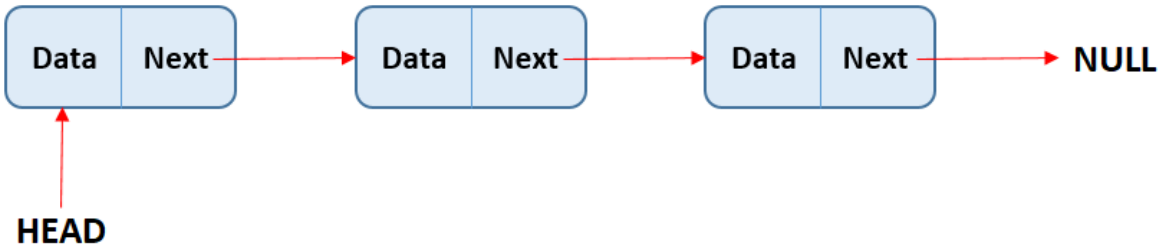
- Singly linked-List

Objective

To write a program that implements the basic operations of the linked list in Java.

Linked List Overview

- ✔ List is a collection of components, called nodes. Every node (except the last node) contains the address of the next node. Every node in a linked list has two components:
  - ➢ One to store the relevant information (that is, data)
  - ➢ One to store the address, called the link or next, of the next node in the list.
- ✔ The first node in the list is called the head or first.



Inserting a Node in a Singly-linked List

The node to be inserted must be first created, after the creation the insertion follows one of the following step based on the position where node is to be inserted.

- **Insertion At the front of the linked list**
- **Insertion After a given node**
- **Insertion At the end of the linked list**

Removing a Node from a Singly-linked List

Following aspects are to be taken into account before removing a node from a linked list:

- **list may be empty**
- **list may contain a single node**
- **list has more than one node**

Note: The Linked List implementation must be Generic. Create a proper menu in Main Class so that the user can perform different functionalities on the Linked List. You can use the Main Class from previous lab and make appropriate changes to it. Keep in mind that all the functions are to be performed on same list so initialize the Linked List carefully in Main Class.

Tasks 1

Create your own class of Linked List, a Node Class and a Main Class.

- Node Class should contain generic data, a reference to next and a parameterized constructor.
- Link-list Class should have head only.

Implement function called **Display( )** to print the elements of complete list

Tasks 2

Add functionalities to your Linked List Class by implementing the methods below:

InsertAtStart( ) to add node at the beginning of list. Create a new node, add it to the list and reassign the head. Make sure the list is not empty.

InsertAtEnd( ) to add node at the end of list. Create a new node, traverse the list to reach list end and add it to the list. Make sure the list is not empty.

InsertAfter( ) to inset new Node after specific Node in list. Make sure that list is not empty. Traverse through the whole list and find the given node and add the new node after that. (Hint use p and c new nodes).

Tasks 3

Add functionalities to your Linked List Class by implementing the methods below:

DeleteFromStart( )to delete node from the beginning of list. Use head Node to delete first Node.

DeleteFromEnd( )to delete node from the last of list. Traverse the whole link-list and delete the last node.

DeleteAny( )to delete any node from the list. Find the node and delete it using p and c Node.

DeleteMax( ) to delete node with maximum value in list. Create a **findMax()** helper method to find maximum and pass the value to **DeletAny()** function.

DeleteMin( ) to delete node with minimum value in list. Create a **findMin()** helper method to find minimum and pass the value to **DeletAny()** function.

Reverse() to reverse the whole link list using p and c Node.