Lab – 04
9/02/2025

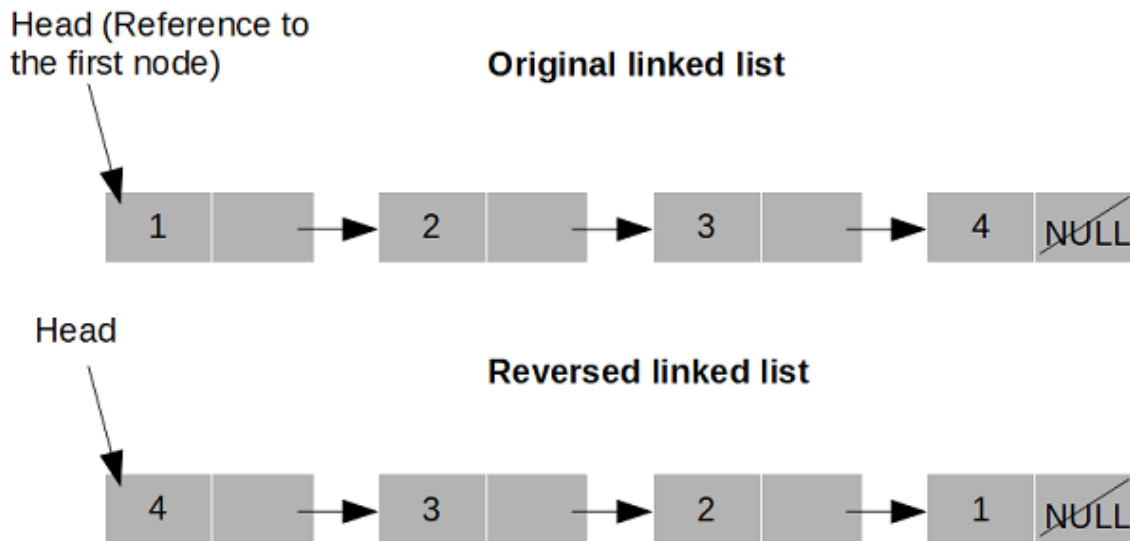## List ADT - Singly Linked List

Note:

1.   Use only Visual Studio code type your program and run your code.

2.   Always follow industry coding best practices.

3.   To compile your file, follow the steps below

- Save your file as a .cpp file
- Go to the location where you have stored the file via terminal
- Compile as "g++ -o objectfilename filename.cpp"
- Run as "./objectfilename"

A. Write a C++ menu-driven program to implement List ADT using a singly linked list. Maintain proper boundary conditions and follow good coding practices. The List ADT has the following operations,

1. Insert Beginning
2. Insert End
3. Insert Position
4. Delete Beginning
5. Delete End
6. Delete Position
7. Search
8. Display
9. Display Reverse
10. Reverse Link
11. Exit

Option 9 does not change the structure of the list. It only prints the contents of the list in reverse order.
Option 10 does the following,

Head (Reference to the first node)

Original linked list

Head

Reversed linked list

What is the time complexity of each of the operations?

B. Write a C++ menu-driven program to implement List ADT using a singly linked list. You have a gethead() private member function that returns the address of the head value of a list. Maintain proper boundary conditions and follow good coding practices. The List ADT has the following operations,

1. Insert Ascending
2. Merge
3. Display
4. Exit

Option 1 inserts a node so the list is always in ascending order. Option 2 takes two lists as input, and merges two lists into a third list. The third list should also be in ascending order. Convert the file into a header file and include it in a C++ file. The second C++ consists of 3 lists and has the following operations,

1. Insert List1
2. Insert List2
3. Merge into List3
4. Display
5. Exit

The time complexity of operation 1 is **O(n),** operation 2 is **O(m)**, and operation 3 should be done in O(**n + m**).