**Department of Computing**

**School of Electrical Engineering and Computer Science**

**CS-250: Data Structure and Algorithms**

**Class: BESE 15AB**

**Lab 3: Implementation of Linked List with its Operations**

**Date: 26th Sep, 2025**

**Time: 10:00am – 01:00 pm & 2:30pm – 5:00pm**

**Lab Engineer: Hadia Tahir**

**Lab 3: Implementation of Linked List with its Operations**

**Introduction**

Students have learned the fundamental concepts of linked lists in the lectures. This lab will introduce students with the practical implementation of a linked list and different operations that can be performed on a linked list.

**Objectives**

Objective of this lab is to get familiar with singly linked list and implement them in C++.

**Tools/Software Requirement**

Visual Studio C++, VS Code

**Helping Material**

Lecture slides, Text book

**Description**

**Singly Linked List**

A Linked List is a data structure consisting of a group of nodes which together represent a sequence. Under the simplest form, each node is composed of two parts i.e. data part and a reference part (also known as, a link) to the next node in the sequence.

**Linked List Structure:**

A linked list is made up of nodes, where each node contains:

**Data:** The value the node stores.

**Pointer:** A reference (pointer) to the next node in the list.

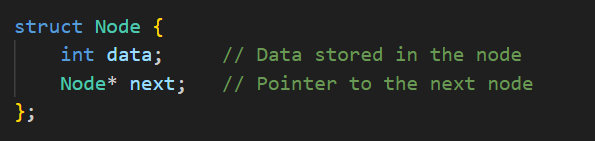
This structure allows efficient insertion or removal of elements from any position in the sequence.



**The basic operation consist of**

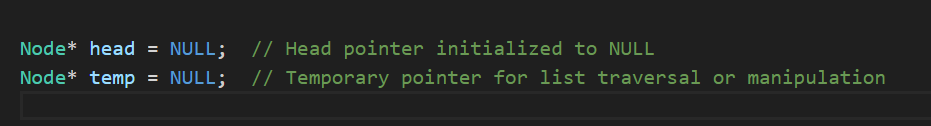
* ***Creating*** the list.

A linked list is created using nodes



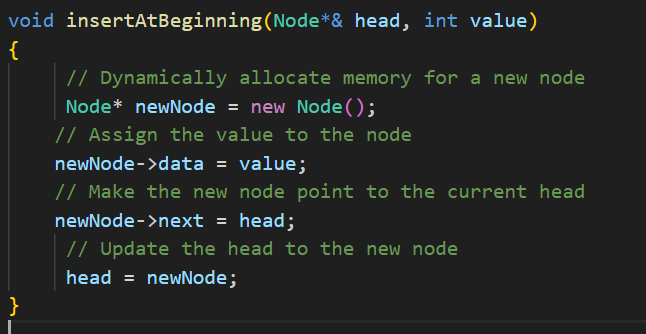
* ***Initialize*** pointers to NULL.

When working with linked lists, pointers must be initialized to NULL when they aren't pointing to any node. To create a linked list, initialize the head (starting node) to NULL to signify that the list is initially empty.

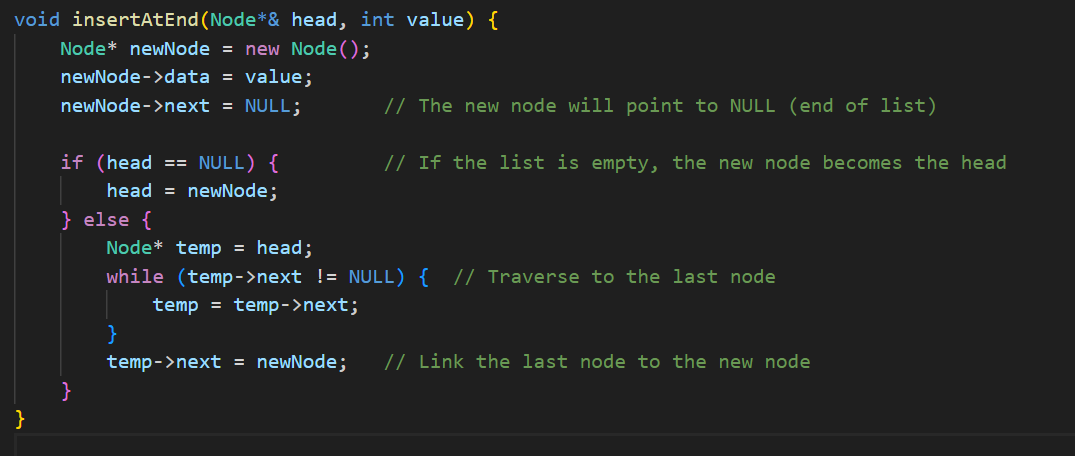


* ***Inserting*** nodes at beginning, last and from a specific location.

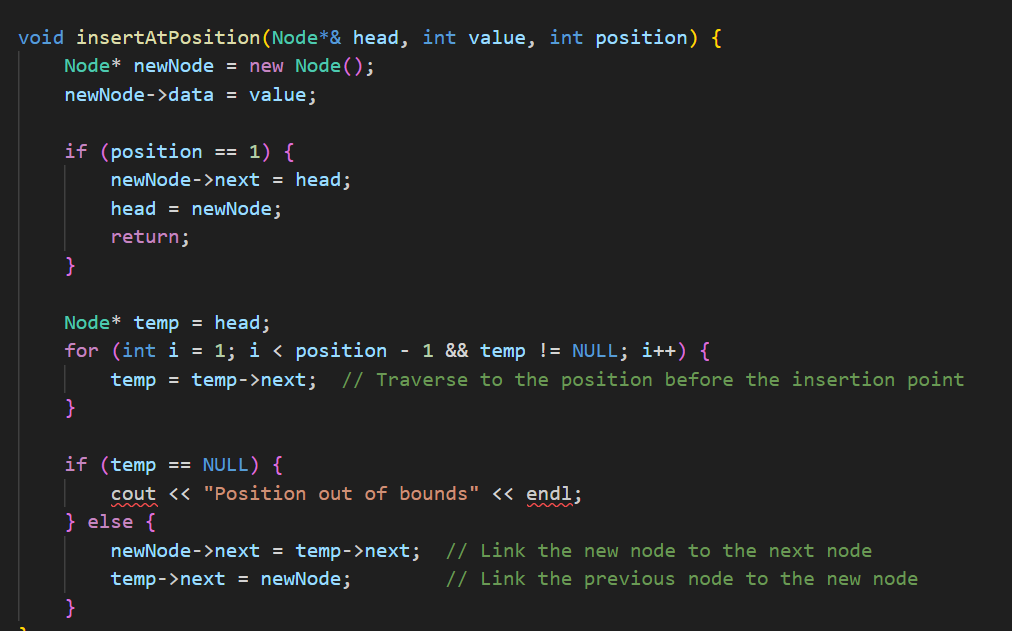
*At the beginining:*



*At the end:*

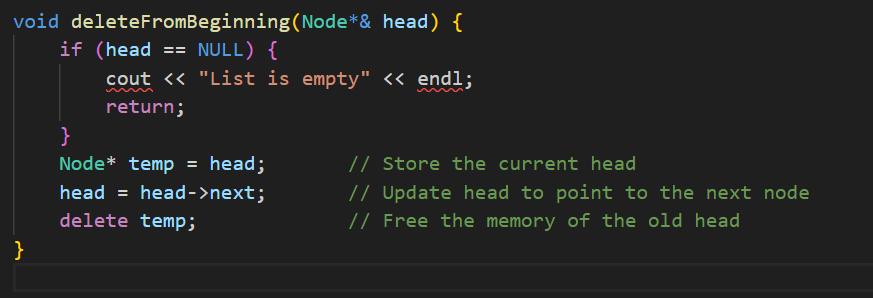


*At the specific position:*

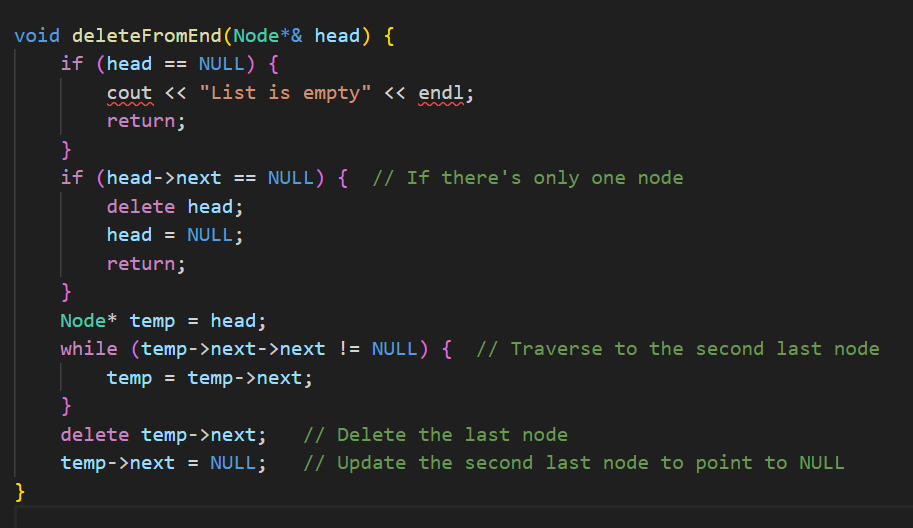


* ***Deletion*** of nodes from beginning, last and from a specific location.

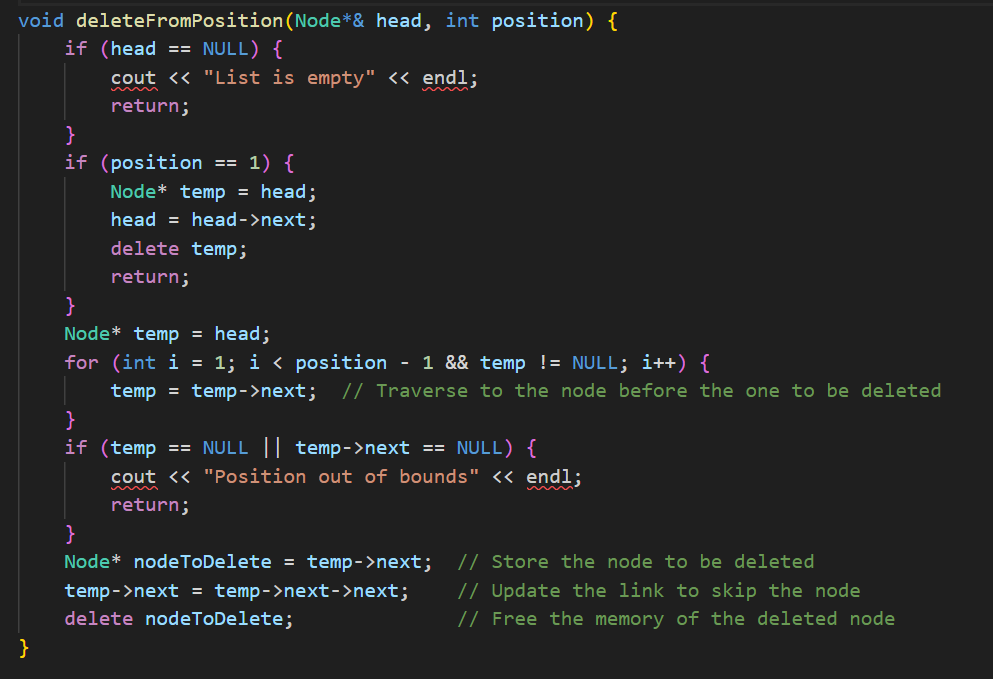
***Delete from Beginning***



***Delete from End***

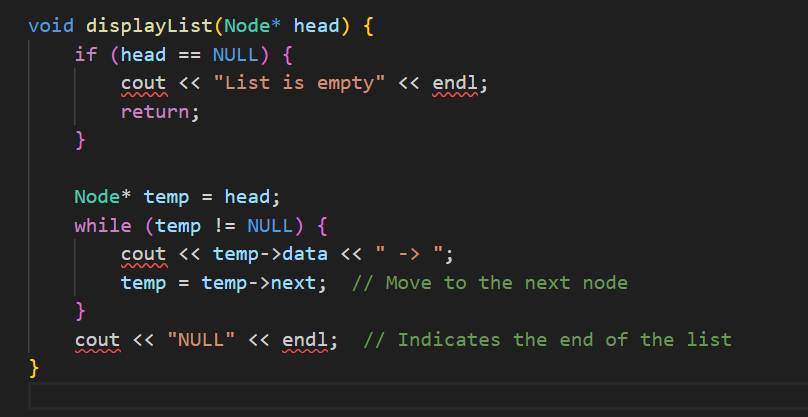


***Delete from a Specific Position***



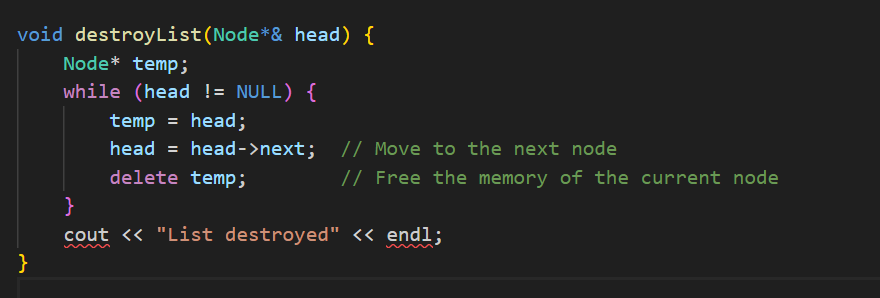
* ***Traversing*** the list.

To display or process all elements in the list, we need to traverse it from the head to the end.



* ***Destroying*** the list.

To completely free the memory used by the list, we can traverse it and delete each node.



**Lab Tasks**

Write a C++ program that can

1. Create a simple linked list using functions to insert nodes at the head.

2. Make a function that can insert another node at 3rd location.

3. Make a function that can display the lists made in 1 and 2.

4. Write a function that can delete node from the linked list selected by the user. Display it as well.

5. Write a function that can count the number of nodes present in list.

6. Create menu in main function to give call to all of the above functions depending upon user’s input.

7. Implement a queue using a singly linked list in C++. The queue should support the following operations:

* enqueue(int value): Insert an element at the end of the queue.
* dequeue(): Remove an element from the front of the queue.
* peek(): Return the value of the front element without removing it.
* isEmpty(): Check if the queue is empty.
* display(): Print the queue elements.

8. Implement a circular linked list in C++ and perform the following operations:

* Insert an element at the end.
* Delete an element from the front.
* Display all elements.
* Detect if the list is circular by checking if the last node points back to the first node.

**Important Note:** Please note that you have to develop your solution in C++ (OOP) i.e. using classes and objects. Solution written in a procedural style will not be accepted.

**Hint:** First you will create the relevant classes, and the functions will belong to the List class.

//class of node

class Node {

public:

int value;

node \*next;

};

Required functions for list class are:

void insertAtHead(int new\_value);

void insertAtLocation(int location,int new\_value)

void delete(int del\_value)

void displayList()

void countList()

**Solution**

|  |
| --- |
| Solution |
| Task 1 Code:  Task 1 Output Screenshot: |

**Deliverables**

Compile a single word document by filling in the solution part and submit this Word file on LMS. Insert the solution/answer in this document. You must show the implementation of the tasks in the designing tool, along with your complete Word document to get your work graded. You must also submit this Word document on the LMS.

**Lab Rubrics**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Assessment** | **Doesn’t meet Expectation**  **(1-2)** | **Meets Expectation**  **(3-4)** | **Exceeds Expectation**  **(5)** | **Marks** |
| **Software Problem Realization**  **(CLO2 – PLO2)** | The Student is unable to understand and **outline** the problem and doesn’t use the relevant method to solve it. | The student requires some guidance to completely comprehend the problem and to **differentiate** the data structure and algorithm comprehensively. | The student fully understands the given problem, is able to **analyze** the relevant method to solve it, and develops a detailed program flow. |  |
| **Software Tool Usage**  **(CLO3 – PLO5)** | The student has no idea on how to use the basic tools of the software. The codes have syntax errors, and parts of the codes are missing. Also, they are unable to **imitate** the required output | The student has a limited command on the basic tools of the software and **operated it under supervision**. The codes are correct in terms of their syntax, however, the program output is not always correct. | The student has full command on various tools available in the software. Furthermore, his/her coding is complete and functional, and the program output is correct. Moreover, they can easily **manipulate the code** to design a particular solution |  |
| **Ethics and Adherence to Laboratory Safety Rules**  **(CLO4 - PLO 8)** | The student does not **behave according to** the professional ethics by following ethical norms applicable to the software industry such as acknowledgement while using publicly available data/ code. Disturbs the lab environment, doesn’t take care of safety measures, and/or isn’t punctual. | The student partially demonstrate their commitment to professional ethics by following ethical norms applicable to the software industry such as referencing and acknowledgement while using publicly available data/ code. **Exhibits** better behavior, works by taking into account the safety measures, and is punctual. | The student clearly **express** the commitment to professional ethics by following ethical norms applicable to the software industry such as referencing and acknowledgement while using publicly available data/ code. Encourages others to maintain lab decorum, and alerts them to follow safety measures. |  |