



## CL-2001 Data Structures Lab # 3

### Objectives:

- Linked List(insertion)
- Linked List(deletion)
- Linked List(searching)
- Linked List(Traversal)

**Note: Carefully read the following instructions (*Each instruction contains a weightage*)**

1. There must be a block of comments at start of every question's code by students; the block should contain brief description about functionality of code.
2. Comment on every function and about its functionality.
3. Mention comments where necessary such as comments with variables, loop, classes etc to increase code understandability.
4. Use understandable name of variables.
5. Proper indentation of code is essential.
6. Write a code in C++ language.
7. Make a Microsoft Word file and paste all of your C++ code with all possible screenshots of every task **outputs in Microsoft Word and submit word file. Do not submit .cpp file.**
8. First think about statement problems and then write/draw your logic on copy.
9. After copy pencil work, code the problem statement on MS Studio C++ compiler.
10. At the end when you done your tasks, attached C++ created files in MS word file and make your submission on Google Classroom. (Make sure your submission is completed).
11. Please submit your file in this format **19F1234\_L4**.
- 12.Do not submit your assignment after deadline. Late and email submission is not accepted.**
- 13.Do not copy code from any source otherwise you will be penalized with negative marks.**

## Problem: 1 | Insertion in Linked list

Write a C++ program to insert values at the beginning, middle and at the end of a Single Linked List.

```
struct node
{
    int data;
    node* next;
};
class LL
{
    node *head = nullptr;
public:
    LL()
    {
        head = nullptr;
        curr = nullptr;
    }
    bool isempty(){}
    node* create_node(int value){}
    void insert_pos(){}
    void display(){}
}
int main()
{
    LL obj;
    cout << "1.Insert Node at beginning" << endl;
    cout << "2.Insert node at last" << endl;
    cout << "3.Insert node at position" << endl;
    cout << "4.Displaying Linked List:" << endl;
}
```

## Problem: 2 | Deletion in Linked list

Write a C++ program to search and delete node of a Single Linked List.

## Problem: 3 | Fix the code

The following program is supposed to allocate nodes, put them into a linked list, print them out and then destroy them. Fix the bugs in the program and rewrite the code. Your program should have complete implementation required to run the following function. Take dummy data execute it, take snap shot and paste it in the document.

```
struct linkedList {
    int number;
    linkedList* next;
};
```

```
void AddNode()
{
    linkedList* head = NULL;
    linkedList* current;
    for (int i = 0; i < 10; i++) {
        current = new linkedList;
        current->number = i;
        current->next = head;
        head = current;
    }
    while (head->next != NULL) {
        cout << head->number << endl;
        current = head->next;
        delete current;
        head = current;
    }
    return 0;
}
```

## Problem: 4 | Fix a code

Write output for the code below, run the program in your compiler take snapshot of output and place it in your document. Explain the working of this code

```
struct myList {
    int data;
    myList* next;
};
int main() {
    myList* head;
    myList* cur;
    myList* previous = NULL;
    for (int i = 0; i < 4; i++) {
        head = new myList;
        head->data = 0;
        head->next = previous;
        for (cur = previous; cur != NULL; cur = cur->next)
            head->data += 1 + 2 * cur->data;
        previous = head;
    }
    while (previous != NULL) {
        cout << previous->data << endl;
        cur = previous;
        previous = previous->next;
        delete cur;
    }
    return 0;
}
```



## Problem: 5 | Concatenating Lists

Write a program that concatenates two linked list objects of characters. The program should include function concatenate, which takes references to both list objects as arguments and concatenates the second list to the first list.



Best of luck

**You are done with your exercise, submit on Classroom at given time.**