Week 7 - Linked Lists

Anh Huynh

1. Create a linked list in C++, add nodes and delete nodes at the start of the list.

Singly Linked-List

```
#include <iostream>
using namespace std;

struct Node{
    int val;                        // holds the value of the node
    Node* next;                     // pointer to the next node in list

    // constructor that sets the node's value and
    // makes sure it doesn't point to anything yet
    Node(int val): val(val), next(nullptr){}
};


Node* deleteFront(Node* head){
    if(head == nullptr){            // check if head is null
        cout << "List is empty.\n";   // means that list is empty
        return nullptr;             // return null
    }

    Node* temp = head;              // set temp node to head
    head = head->next;              // point head to next node
    delete temp;                    // delete old head node
    return head;                    // return the new head node
}

void printList(Node* head){
    Node* curr = head;              // start from the head node

    while (curr != nullptr){        // loop until the end
        cout << curr->val << " - "; // print value of node
        curr = curr->next;          // move to the next node
    }
    cout << "NULL\n";               // mark end of list
}

int main(){
    // creating the nodes
    Node* head = new Node(0);
    Node* first = new Node(1);
    Node* second = new Node(2);
```

```
    Node* third = new Node(3);

    // linking the nodes
    head->next = first;
    first->next = second;
    second->next = third;
    third->next = nullptr;

    // print the original list
    cout << "\nOriginal List: \n";
    printList(head);

    // delete nodes from front one by one
    // by using the deleteFront function
    cout << "\nDeleting front nodes:\n";
    head = deleteFront(head);
    printList(head);

    head = deleteFront(head);
    printList(head);

    head = deleteFront(head);
    printList(head);

    return 0;   // program finishes

}
```

Video Link: