ASSIGNMENT – WEEK2

Implement a Linked List in Python Using OOP and Delete the Nth Node

- Create a Python program that implements a singly linked list using Object-Oriented Programming (OOP) principles. Your implementation should include the following: A Node class to represent each node in the list. A LinkedList class to manage the nodes, with methods to: Add a node to the end of the list Print the list Delete the nth node (where n is a 1-based index) Include exception handling to manage edge cases such as: Deleting a node from an empty list Deleting a node with an index out of range Test your implementation with at least one sample list.

```
class Node:
    def __init__(self, data):
        self.data = data
        self.next = None


class LinkedList:
    def __init__(self):
        self.head = None

    def add_node(self, data):
        new_node = Node(data)
        if self.head is None:
            self.head = new_node
            return
```

```
        temp = self.head
        while temp.next:
            temp = temp.next
        temp.next = new_node

    def print_list(self):
        """Print the entire list."""
        if self.head is None:
            print("List is empty.")
            return
        temp = self.head
        while temp:
            print(temp.data, end=" -> ")
            temp = temp.next
        print("None")

    def delete_nth_node(self, n):
        if self.head is None:
            raise Exception("Cannot delete from an empty list.")

        if n <= 0:
            raise ValueError("Index should be 1 or greater.")

        # Deleting the head node
        if n == 1:
            deleted_data = self.head.data
```

```
            self.head = self.head.next
            print(f"Deleted node with data: {deleted_data}")
            return

        temp = self.head
        count = 1
        while temp is not None and count < n - 1:
            temp = temp.next
            count += 1

        if temp is None or temp.next is None:
            raise IndexError("Index out of range.")

        deleted_data = temp.next.data
        temp.next = temp.next.next
        print(f"Deleted node with data: {deleted_data}")


#  Sample Test Code

if __name__ == "__main__":
    ll = LinkedList()

    # Add nodes
    ll.add_node(10)
    ll.add_node(20)
```

```
ll.add_node(30)
ll.add_node(40)

print("Initial list:")
ll.print_list()

try:
    ll.delete_nth_node(2)
except Exception as e:
    print("Error:", e)

print("After deleting 2nd node:")
ll.print_list()

# Try deleting out of range
try:
    ll.delete_nth_node(10)
except Exception as e:
    print("Error:", e)

# Try deleting from empty list
empty_list = LinkedList()
try:
    empty_list.delete_nth_node(1)
except Exception as e:
    print("Error:", e)
```

OUTPUT:

```
Initial list:
10 -> 20 -> 30 -> 40 -> None
Deleted node with data: 20
After deleting 2nd node:
10 -> 30 -> 40 -> None
Error: Index out of range.
Error: Cannot delete from an empty list.
[Finished in 369ms]
```