Assignment 2:

- **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 not self.head:
            self.head = new_node
        else:
            current = self.head
            while current.next:
                current = current.next
            current.next = new_node

    def print_list(self):
        current = self.head
        if not current:
            print("List is empty.")
            return
        while current:
            print(current.data, end=" -> ")
            current = current.next
        print("None")

    def delete_nth_node(self, n):
        if not self.head:
            raise Exception("Cannot delete from an empty list.")

        if n <= 0:
            raise ValueError("Index must be a positive integer.")

        if n == 1:
            print(f"Deleting node with data: {self.head.data}")
            self.head = self.head.next
            return

        current = self.head
        prev = None
        count = 1

        while current and count < n:
            prev = current
            current = current.next
            count += 1

        if not current:
            raise IndexError("Index out of range.")

        print(f"Deleting node with data: {current.data}")
        prev.next = current.next

def main():
    ll = LinkedList()

    print("=== Singly Linked List Program ===")

    while True:
        print("\nMenu:")
        print("1. Add a node")
        print("2. Print the linked list")
        print("3. Delete the nth node")
        print("4. Exit")

        choice = input("Enter your choice (1-4): ")

        if choice == '1':
            try:
                data = int(input("Enter data to add: "))
                ll.add_node(data)
                print("Node added successfully.")
            except ValueError:
                print("Please enter a valid integer.")

        elif choice == '2':
            print("Linked list:")
            ll.print_list()

        elif choice == '3':
            try:
                n = int(input("Enter the position (1-based index) of the node to delete: "))
            ll.delete_nth_node(n)
         except Exception as e:
            print("Error:", e)

      elif choice == '4':
         print("Exiting program. Goodbye!")
         break

      else:
         print("Invalid choice. Please enter 1, 2, 3, or 4.")

if __name__ == "__main__":
   main()
```

Output:

```
=== Singly Linked List Program ===

Menu:
1. Add a node
2. Print the linked list
3. Delete the nth node
4. Exit
Enter your choice (1-4): 1
Enter data to add: 10
Node added successfully.

Menu:
1. Add a node
2. Print the linked list
3. Delete the nth node
4. Exit
Enter your choice (1-4): 1
Enter data to add: 20
Node added successfully.

Menu:
1. Add a node
2. Print the linked list
3. Delete the nth node
4. Exit
Enter your choice (1-4): 1
Enter data to add: 30
Node added successfully.

Menu:
1. Add a node
```

```
2. Print the linked list
3. Delete the nth node
4. Exit
Enter your choice (1-4): 1
Enter data to add: 40
Node added successfully.

Menu:
1. Add a node
2. Print the linked list
3. Delete the nth node
4. Exit
Enter your choice (1-4): 50
Invalid choice. Please enter 1, 2, 3, or 4.

Menu:
1. Add a node
2. Print the linked list
3. Delete the nth node
4. Exit
Enter your choice (1-4): 2
Linked list:
10 -> 20 -> 30 -> 40 -> None

Menu:
1. Add a node
2. Print the linked list
3. Delete the nth node
4. Exit
Enter your choice (1-4): 3
Enter the position (1-based index) of the node to delete: 2
Deleting node with data: 20

Menu:
1. Add a node
2. Print the linked list
3. Delete the nth node
4. Exit
Enter your choice (1-4): 2
Linked list:
10 -> 30 -> 40 -> None

Menu:
1. Add a node
2. Print the linked list
3. Delete the nth node
4. Exit
Enter your choice (1-4): 3
Enter the position (1-based index) of the node to delete: 1
Deleting node with data: 10

Menu:
1. Add a node
2. Print the linked list
3. Delete the nth node
4. Exit
Enter your choice (1-4): 3
Enter the position (1-based index) of the node to delete: 4
Error: Index out of range.

Menu:
1. Add a node
2. Print the linked list
3. Delete the nth node
4. Exit
Enter your choice (1-4): 2
Linked list:
30 -> 40 -> None

Menu:
1. Add a node
2. Print the linked list
3. Delete the nth node
4. Exit
Enter your choice (1-4): 3
Enter the position (1-based index) of the node to delete: 2
Deleting node with data: 40

Menu:
1. Add a node
2. Print the linked list
3. Delete the nth node
4. Exit
Enter your choice (1-4): 3
Enter the position (1-based index) of the node to delete: 1
Deleting node with data: 30

Menu:
1. Add a node
2. Print the linked list
3. Delete the nth node
4. Exit
Enter your choice (1-4): 2
Linked list:
List is empty.
```

```
Menu:
1. Add a node
2. Print the linked list
3. Delete the nth node
4. Exit
Enter your choice (1-4): 3
Enter the position (1-based index) of the node to delete: 3
Error: Cannot delete from an empty list.

Menu:
1. Add a node
2. Print the linked list
3. Delete the nth node
4. Exit
Enter your choice (1-4): 4
Exiting program. Goodbye!
```