Explanation of Linked Lists with Visual Representation and Code

A linked list is a data structure where each element (node) contains:

1. Data: The value stored in the node.
2. Pointer/Reference: A reference to the next node in the sequence.

Types of Linked Lists

1. Singly Linked List: Each node points to the next node.
2. Doubly Linked List: Each node points to both its previous and next nodes.

How Linked Lists Work

Here is a visual representation of a singly linked list:

```
[Data: 10] -> [Data: 20] -> [Data: 30] -> null
```

Each node contains a value (Data) and a reference (Next) to the next node.

C# Code Example: Singly Linked List

```
using System;
using System.Collections.Generic;

class Node
{
    public int Data { get; set; }
    public Node Next { get; set; }
    public Node(int data)
    {
        Data = data;
        Next = null;
    }
}

class LinkedList
{
    private Node head;

    // Add a new node to the end
    public void AddLast(int data)
    {
        Node newNode = new Node(data);
        if (head == null)
        {
            head = newNode;
        }
        else
        {
            Node current = head;
            while (current.Next != null)
            {
                current = current.Next;
            }
            current.Next = newNode;
        }
    }

    // Display the list
    public void PrintList()
    {
        Node current = head;
        while (current != null)
        {
            Console.Write($"{current.Data} -> ");
            current = current.Next;
        }
        Console.WriteLine("null");
    }

    // Delete a node with specific value
    public void Delete(int data)
    {
        if (head == null) return;

        if (head.Data == data)
        {
            head = head.Next;
            return;
        }
        Node current = head;
        while (current.Next != null && current.Next.Data != data)
        {
            current = current.Next;
        }

        if (current.Next != null)
        {
            current.Next = current.Next.Next;
        }
    }
}

class Program
{
    static void Main(string[] args)
    {
        LinkedList list = new LinkedList();
        list.AddLast(10);
        list.AddLast(20);
        list.AddLast(30);

        Console.WriteLine("Initial List:");
        list.PrintList();

        list.Delete(20);
        Console.WriteLine("After Deleting 20:");
        list.PrintList();
    }
}
```

Visual Representation of Operations

1. Adding Nodes:

   Add 10:

   [10] -> null

   Add 20:

   [10] -> [20] -> null

   Add 30:

   [10] -> [20] -> [30] -> null

2. Deleting a Node (e.g., 20):

   Before Deletion:

   [10] -> [20] -> [30] -> null

   After Deletion:

   [10] -> [30] -> null