

# LINKED LIST

Create a linked list using C++ program and insert the elements in linked list.  
Also display the elements of linked list.

## Instruction: -

- Create a structure named Node
- Create a class named LinkedList
  - Create constructor of class
  - Create a insert and display member function.

## C++ Code

```
1. #include <iostream>
2. using namespace std;
3.
4. // Node structure
5. struct Node
6. {
7.     int data;
8.     Node *next;
9. };
10.
11. // Linked List class
12. class LinkedList
13. {
14. public:
15.     Node *head;
16.
17.     LinkedList()
18.     {
19.         head = nullptr;
20.     }
21.
22.     // Insert at the end
23.     void insert(int value)
24.     {
25.         Node* newNode = new Node();
26.         newNode->data = value;
27.         newNode->next = nullptr;
28.
29.         if (head == nullptr)
30.         {
31.             head = newNode;
32.             return;
33.         }
34.
35.         Node* temp = head;
36.         while (temp->next != nullptr)
37.         {
38.             temp = temp->next;
39.         }
40.
41.         temp->next = newNode;
42.     }
43.
44.     void display()
45.     {
46.         Node* temp = head;
47.         while (temp != nullptr)
48.         {
49.             cout << temp->data << " ";
50.             temp = temp->next;
51.         }
52.     }
53.
54.     ~LinkedList()
55.     {
56.         Node* temp = head;
57.         while (temp != nullptr)
58.         {
59.             Node* nextNode = temp->next;
60.             delete temp;
61.             temp = nextNode;
62.         }
63.     }
64. }
```

```

39.         }
40.         temp->next = newNode;
41.     }
42.
43. // Display the list
44. void display()
45. {
46.     Node* temp = head;
47.     while (temp != nullptr)
48.     {
49.         cout << temp->data << " ";
50.         temp = temp->next;
51.     }
52.     cout << endl;
53. }
54. };
55.
56. int main()
57. {
58.     LinkedList list;
59.     list.insert(10);
60.     list.insert(20);
61.     list.insert(30);
62.     list.insert(40);
63.
64.     list.display();
65.
66.     return 0;
67. }
```

### Explanation: -

1. Node structure that holds:
  - data: an integer to store the value.
  - next: a pointer to the next node in the list.
2. head is a pointer that stores the address of the first node.
3. The constructor initializes head to nullptr, meaning the list is initially empty.
4. A new node is created dynamically using new.