

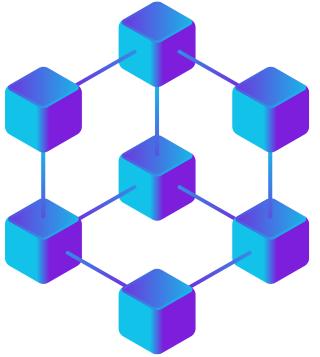
ساختمان داده:

# لیست پیوندی



@ShayanAryania

toooooooooo>>

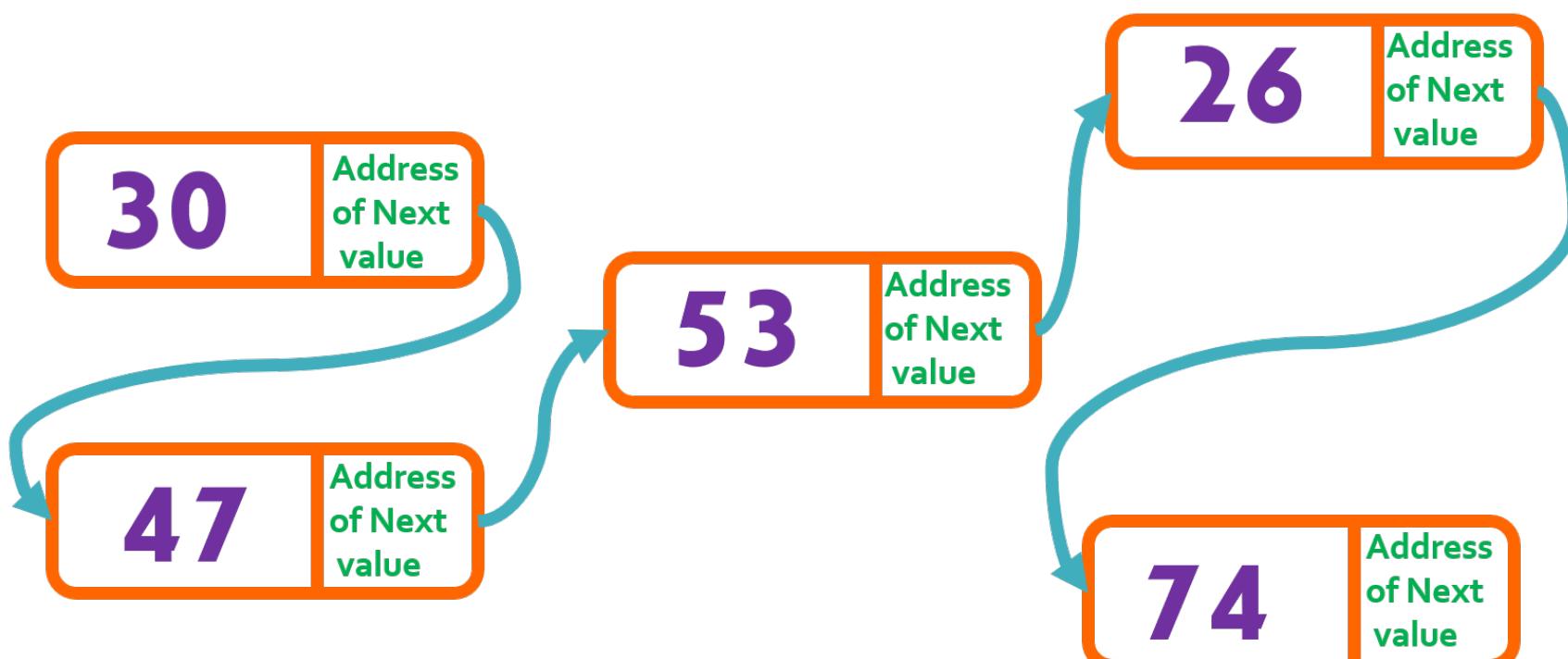


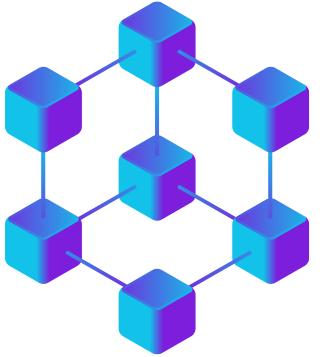
# لیست پیوندی

Linked List

ساختاری شامل دنباله‌ای از عناصر است که هر عنصر دارای اشاره‌گری به عنصر بعدی در دنباله است.

لیست پیوندی از جهاتی شبیه به آرایه است ولی مزیت مهم فهرست پیوندی نسبت به آرایه‌ها این است که ترتیب قرار گرفتن داده‌ها در آن با ترتیب قرار گرفتن آن‌ها در حافظه متفاوت است.





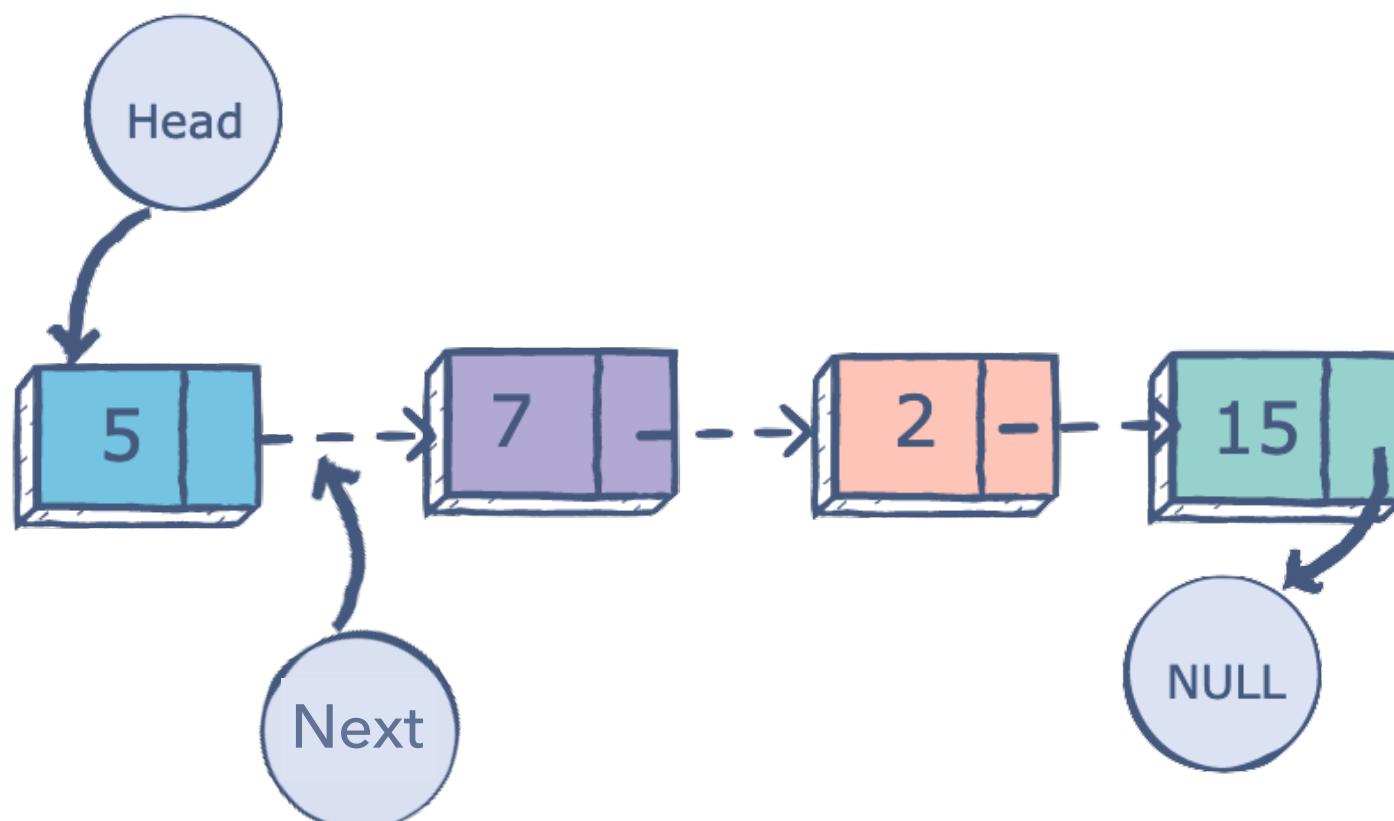
همان طور که در شکل مشخص است، عناصر زیر در یک لیست پیوندی مهم هستند:

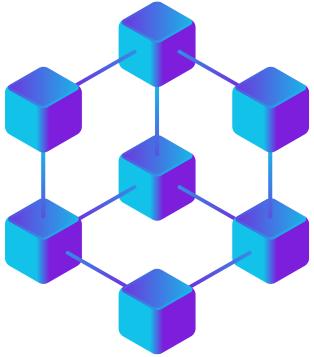
لیست پیوندی شامل یک عنصر پیوندی است که **شروع (Head)** نامیده می شود.

هر لینک، فیلد های داده ای و همچنین یک فیلد پیوند دارد که **بعدی (Next)** نامیده می شود.

هر لینک با استفاده از پیوند Next، به پیوند بعدی خود مربوط می شود.

هر لینک که یک پیوند NULL داشته باشد به معنی انتهای لیست است.





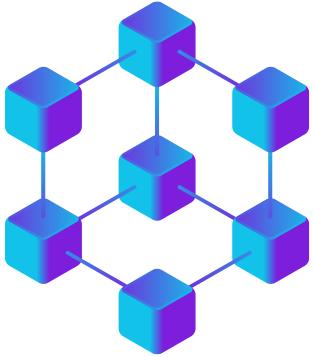
## نکات :

فهرست پیوندی دارای این ویژگی است که درج و حذف گره‌ها در هر نقطه‌ای از فهرست، با تعداد ثابتی از عملیات امکان‌پذیر است.

در این نوع ساختمان داده برای دسترسی به هر عنصر زمان بیشتری (نسبت به آرایه) صرف می‌شود.

حداکثر تعداد لیست پیوندی که می‌توان ایجاد کرد، بستگی به مقدار حافظه سیستم دارد.





# کلیت پیاده سازی لیست پیوندی در C++

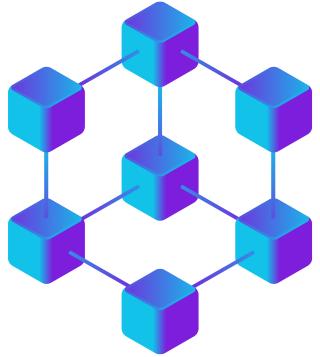
در این کد، لیست را با مجموعه ایی از **Node**‌ها پایه ریزی کرده ایم،

که هر **Node** دارای ۲ بخش است.

بخشی دارای اطلاعات و بخش دیگر دارای اشاره گری به **Node** بعدی است.

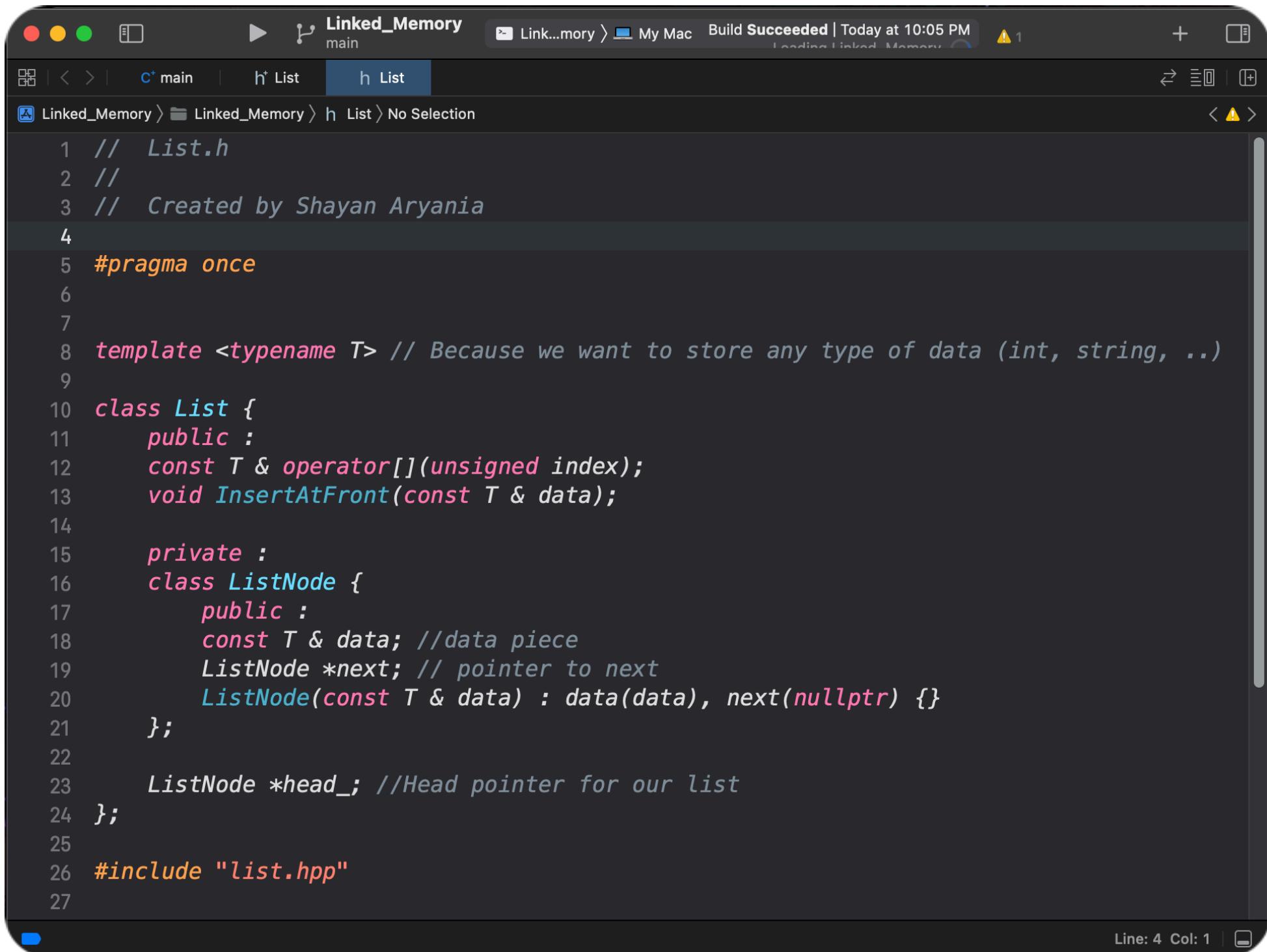
```
template <typename T> // Because we want to store any type of data
    (int, string, ..)
class ListNode {
public :
    T & data; //data piece
    ListNode *next; // pointer to next
    ListNode (T & data) : data(data), next(nullptr) {}
};
```





# C++

## List.h



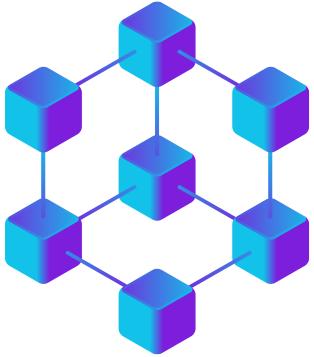
The screenshot shows the Xcode IDE interface with the project navigation bar at the top. The current file is `List.h`, which is selected in the sidebar. The code editor displays the following C++ code:

```
1 // List.h
2 //
3 // Created by Shayan Aryania
4
5 #pragma once
6
7
8 template <typename T> // Because we want to store any type of data (int, string, ..)
9
10 class List {
11     public :
12         const T & operator[](unsigned index);
13         void InsertAtFront(const T & data);
14
15     private :
16         class ListNode {
17             public :
18                 const T & data; //data piece
19                 ListNode *next; // pointer to next
20                 ListNode(const T & data) : data(data), next(nullptr) {}
21         };
22
23         ListNode *head_; //Head pointer for our list
24     };
25
26 #include "list.hpp"
27
```

The code defines a template class `List` for storing any type of data. It includes a public `operator[]` for indexed access and a `InsertAtFront` method for inserting elements at the beginning. A private nested class `ListNode` represents each node in the list, containing a data member and a `next` pointer to the next node. The `head_` pointer points to the start of the list.

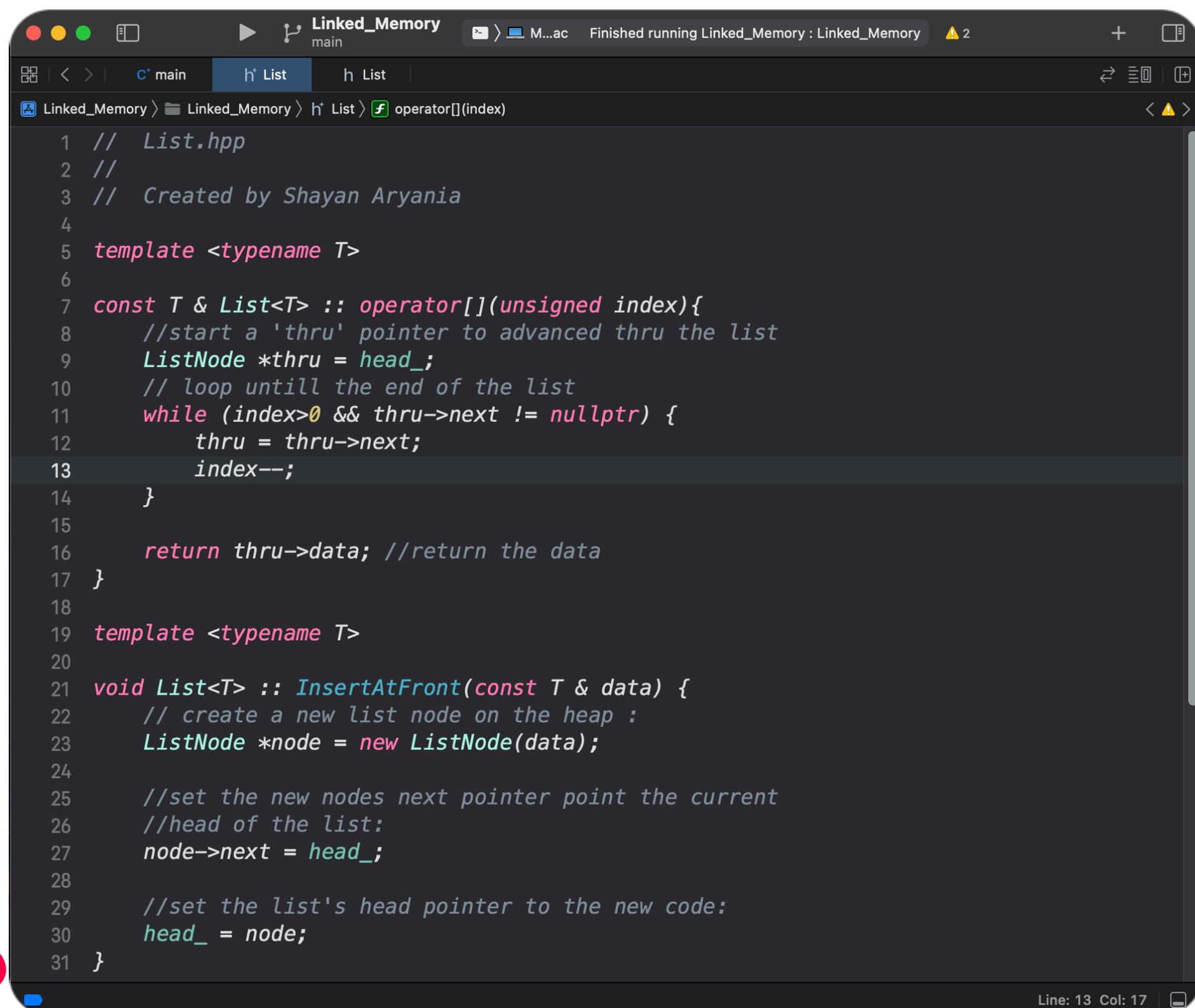


oooooo|oooo>>



# C++

## List.hpp

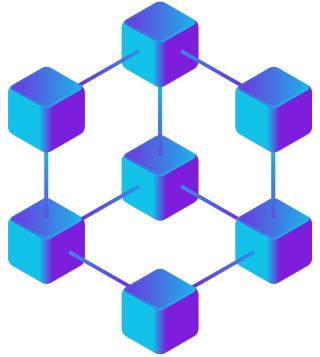


The screenshot shows the Xcode IDE with the project "Linked\_Memory" open. The main window displays the code for "List.hpp". The code implements a linked list with templates for type T. It includes an operator[] function to access elements by index and an InsertAtFront function to add new nodes at the beginning of the list. The code uses a ListNode structure and pointer arithmetic.

```
1 // List.hpp
2 //
3 // Created by Shayan Aryania
4
5 template <typename T>
6
7 const T & List<T> :: operator[](unsigned index){
8     //start a 'thru' pointer to advanced thru the list
9     ListNode *thru = head_;
10    // loop until the end of the list
11    while (index>0 && thru->next != nullptr) {
12        thru = thru->next;
13        index--;
14    }
15
16    return thru->data; //return the data
17 }
18
19 template <typename T>
20
21 void List<T> :: InsertAtFront(const T & data) {
22     // create a new list node on the heap :
23     ListNode *node = new ListNode(data);
24
25     //set the new nodes next pointer point the current
26     //head of the list:
27     node->next = head_;
28
29     //set the list's head pointer to the new code:
30     head_ = node;
31 }
```



000000+000>>



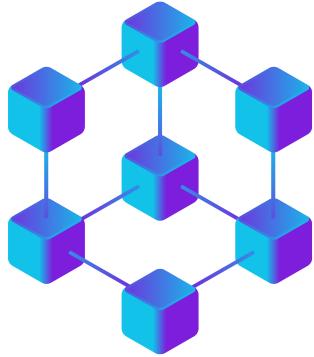
# C++

## Main.cpp

```
Linked_Memory main M...ac Finished running Linked_Memory : Linked_Memory 2
Linked_Memory > main > List > List > main(argc, argv)
1 // main.cpp
2 // Linked_List
3 //
4 // Created by Shayan Aryania
5 //
6
7 #include <iostream>
8 #include "list.h"
9 using namespace std;
10
11 int main(int argc, const char * argv[]) {
12     // insert code here...
13     List<int> list;
14     list.InsertAtFront(3);
15     list.InsertAtFront(2);
16     list.InsertAtFront(1);
17     cout<< " list[0]: " << list[0] << endl;
18     cout<< " list[1]: " << list[1] << endl;
19     cout<< " list[2]: " << list[2] << endl;
20 }
21
list[0]: 1
list[1]: 2
list[2]: 3
Program ended with exit code: 0
```



00000000oo>>



# Python

The screenshot shows a split-screen view of a Python script named `LinkedList.py` in VS Code. The left pane contains the class definition for a linked list, and the right pane contains the execution logic.

```
1 # Node class
2 class Node:
3
4     # Function to initialise the node object
5     def __init__(self, data):
6         self.data = data # Assign data
7         self.next = None # Initialize next as null
8
9
10 # Linked List class contains a Node object
11 class LinkedList:
12
13     # Function to initialize head
14     def __init__(self):
15         self.head = None
16
17     # This function prints contents of linked list
18     # starting from head
19     def printList(self):
20         temp = self.head
21         while (temp):
22             print (temp.data)
23             temp = temp.next
24
25
```

```
26 # Code execution starts here
27 if __name__=='__main__':
28
29     # Start with the empty list
30     llist = LinkedList()
31     llist.head = Node(1)
32     second = Node(2)
33     third = Node(3)
34
35     llist.head.next = second; # Link first node with second
36     second.next = third; # Link second node with the third node
37
38     llist.printList()
```

The terminal below shows the output of running the script:

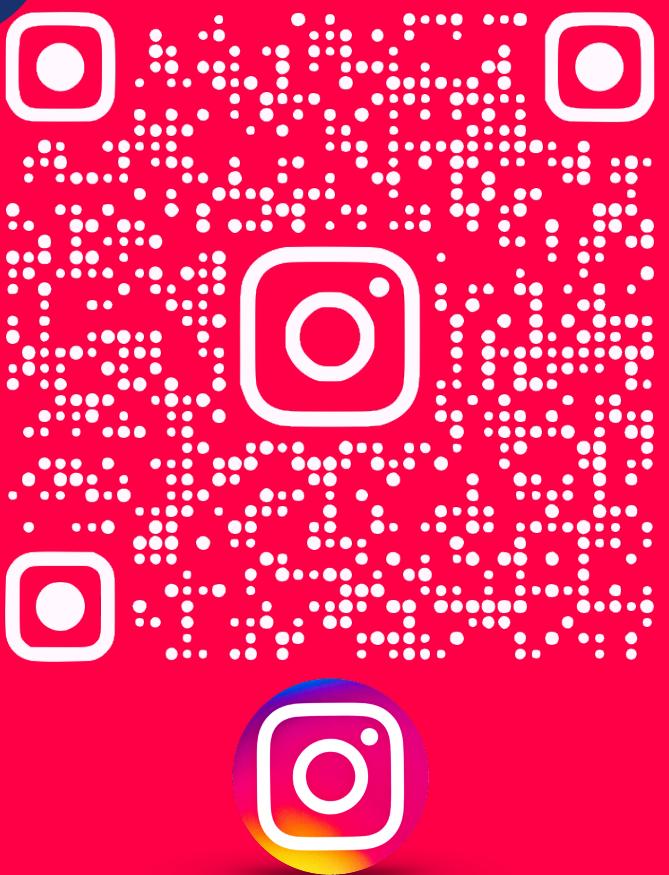
```
shayan@Shayans-MBP Desktop % /usr/bin/env /usr/local/bin/python3.10 /Users/shayan/.vscode/extensions/ms-python.python-2022.0.1814523869/pythonFile
s/lib/python/debugpy/launcher 51090 -- /Users/shayan/Desktop/LinkedList.py
1
2
3
shayan@Shayans-MBP Desktop %
```



00000000+o>>



Follow



[www.instagram.com/shayanaryania](https://www.instagram.com/shayanaryania)



[www.linkedin.com/in/shayanaryania](https://www.linkedin.com/in/shayanaryania)



Like & Share

Save