

Array and Linked List

A decorative graphic in the background consists of several overlapping blue circles and rectangles of varying sizes, creating a sense of depth and motion.

An array is a collection of elements stored in contiguous memory. It allows fast access but has a fixed size.

Array

1

1x23

2

1x24

3

1x25

4

1x26

5

1x27

NULL

1x28

Edit Selection View Go Run Terminal Help ← → 🔍 Introduction-to-Data-Structure 📁

array.cpp U X

```
code > array.cpp > main()
1 #include <iostream>
2 using namespace std;
3
4 int main(){
5     int arr[5] = {1,2,3,4,5};
6
7     for(int i = 0; i < 5;i++){
8         cout << "location: " << &arr[i] << " value: " << arr[i] << endl;
9     }
10    cout << "size of array: " << sizeof(arr) << endl;
11    cout << "size of element: " << sizeof(arr[0]) << endl;
12    cout << "length of array: " << sizeof(arr)/sizeof(arr[0]) << endl;
13    cout << "arr[6]: " << arr[6] << endl;
14    cout << "location of arr[6]: " << &arr[6] << endl;
15 }
```

PROBLEMS DEBUG CONSOLE OUTPUT PORTS SERIAL MONITOR COMMENTS TERMINAL

powershell - code + ⌂ ⌂ ⌂ ⌂

```
PS D:\Documents\GitHub\Introduction-to-Data-Structure> cd code
PS D:\Documents\GitHub\Introduction-to-Data-Structure\code> g++ array.cpp -o array
PS D:\Documents\GitHub\Introduction-to-Data-Structure\code> ./array.exe
location: 0x61fee8 value: 1
location: 0x61feec value: 2
location: 0x61fef0 value: 3
location: 0x61fef4 value: 4
location: 0x61fef8 value: 5
size of array: 20
size of element: 4
length of array: 5
arr[6]: 1987873821
location of arr[6]:0x61ff00
PS D:\Documents\GitHub\Introduction-to-Data-Structure\code>
```

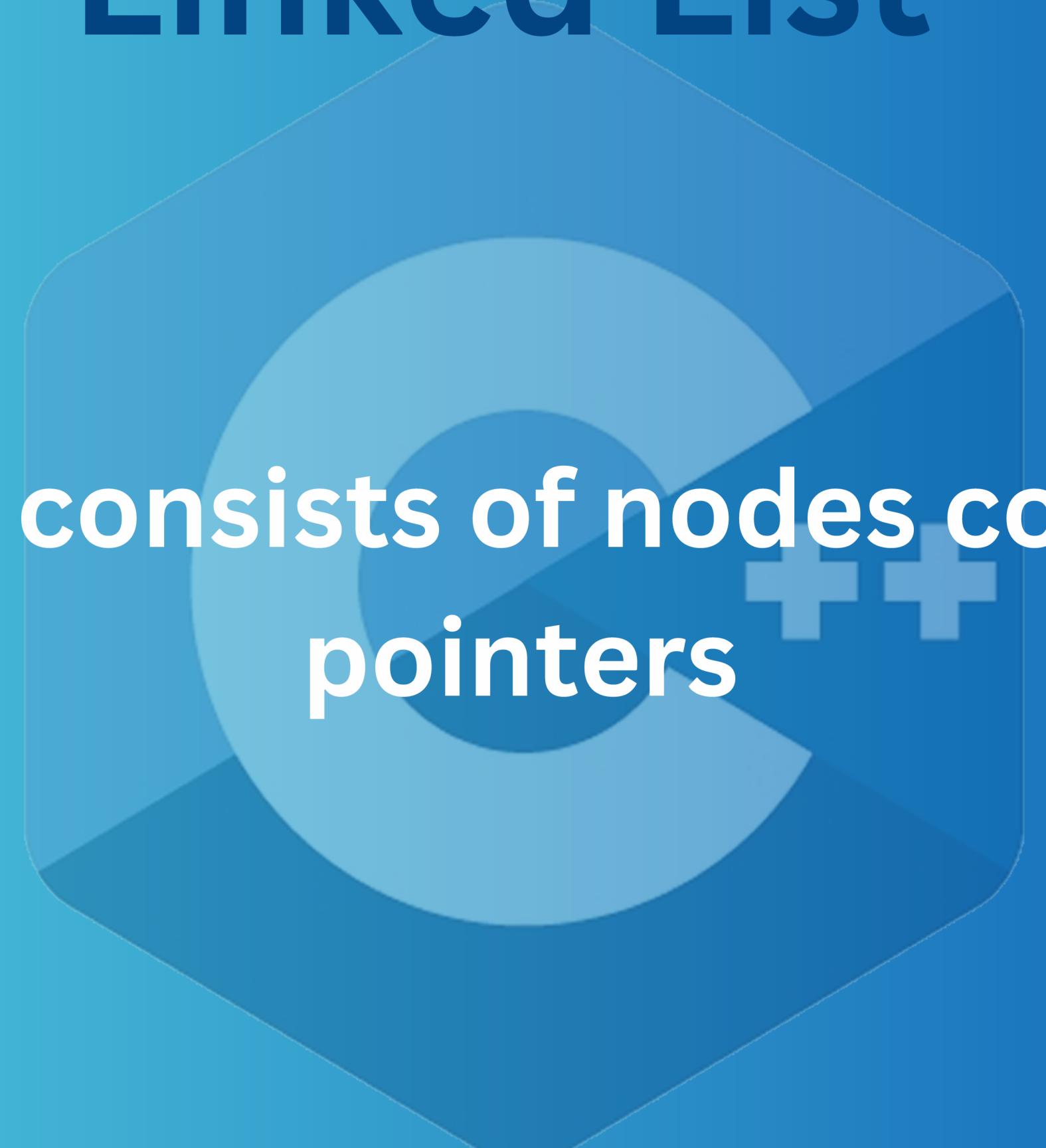
```
int main(){
    int arr[5] = {1,2,3,4,5};
```

```
for(int i = 0; i < 5;i++){
    cout << "location: " << &arr[i] << " value: " << arr[i] << endl;
}
```

```
cout << "size of array: " << sizeof(arr) << endl;
cout << "size of element: " << sizeof(arr[0]) << endl;
cout << "length of array: " << sizeof(arr)/sizeof(arr[0]) << endl;
cout << "arr[6]: " << arr[6] << endl;
cout << "location of arr[6]: " << &arr[6] << endl;
```

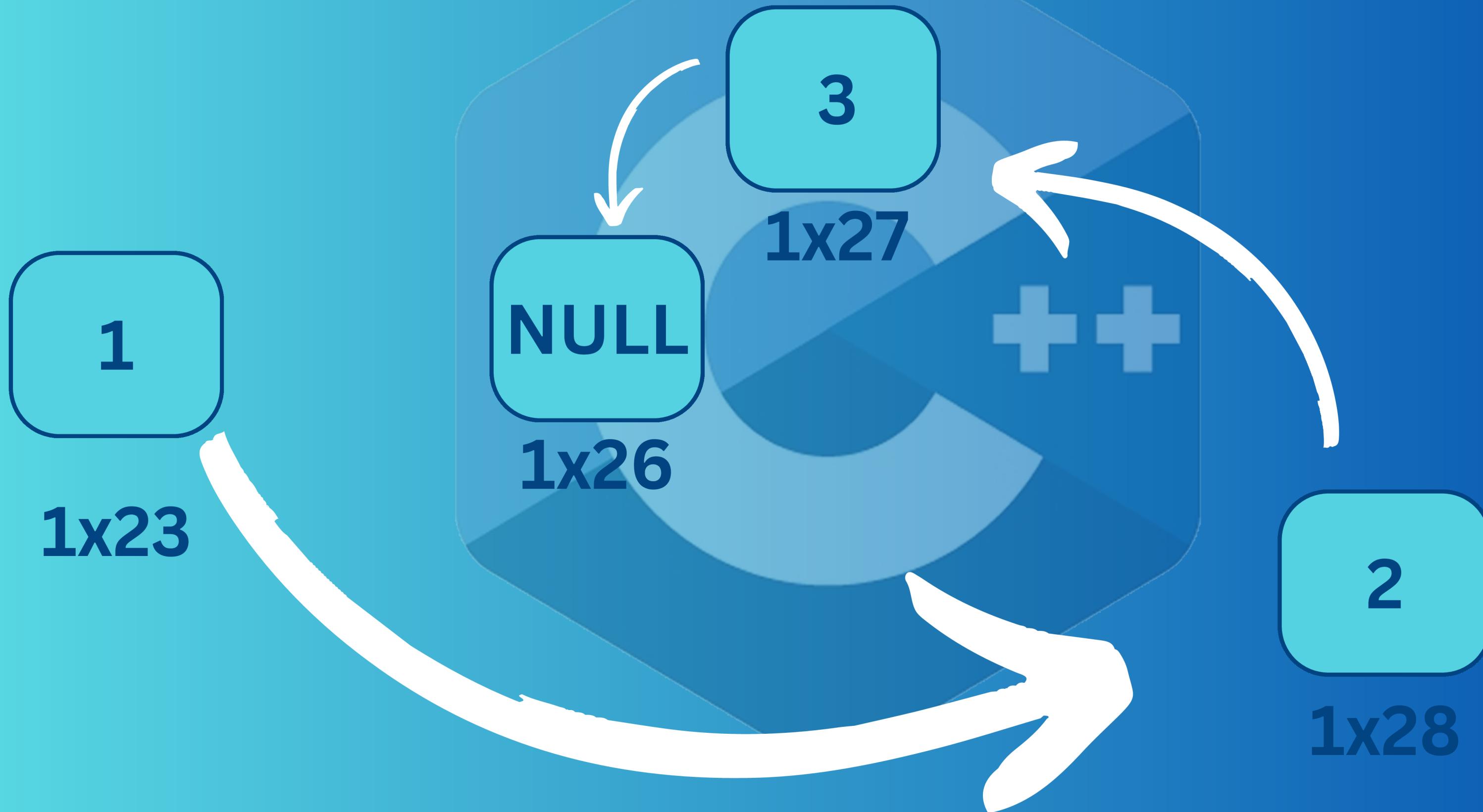
```
location: 0x61fee8 value: 1
location: 0x61feec value: 2
location: 0x61fef0 value: 3
location: 0x61fef4 value: 4
location: 0x61fef8 value: 5
size of array: 20
size of element: 4
length of array: 5
arr[6]: 1987873821
location of arr[6]:0x61ff00
```

Linked List



A linked list consists of nodes connected by pointers

Linked List



Linked List



Edit Selection View Go Run Terminal Help ← → 🔍 Introduction-to-Data-Structure 🌐

array.cpp U linkedlist.cpp U X

code > linkedlist.cpp > main()

```
4 struct Node{  
5     int data;  
6     Node* next;  
7 };  
8  
9 int main(){  
10    Node* head = new Node();  
11    Node* second = new Node();  
12    Node* third = new Node();  
13  
14    head -> data = 1;  
15    head -> next = second;  
16    second -> data = 2;  
17    second -> next = third;  
18    third -> data = 3;  
19    third -> next = NULL;  
20  
21    Node* t = head;  
22    while ( t != NULL){  
23        cout << "location: " << t << "value: " << t -> data << "next: " << t -> next << endl;  
24        t = t -> next;  
25    }  
26 }
```

PROBLEMS DEBUG CONSOLE OUTPUT PORTS SERIAL MONITOR COMMENTS TERMINAL

powershell - code + ×

```
PS D:\Documents\GitHub\Introduction-to-Data-Structure\code> ./linkedlist.exe  
location: 0x10f1be0value: 1 next: 0x10f1bf0  
location: 0x10f1bf0value: 2 next: 0x10f1c00  
location: 0x10f1c00value: 3 next: 0  
PS D:\Documents\GitHub\Introduction-to-Data-Structure\code>
```

```
3
4 struct Node{
5     int data;
6     Node* next;
7 };
8
```

```
int main(){
    Node* head = new Node();
    Node* second = new Node();
    Node* third = new Node();

    head -> data = 1;
    head -> next = second;
    second -> data = 2;
    second -> next = third;
    third -> data = 3;
    third -> next = NULL;
```

```
Node* t = head;
while ( t != NULL){
    cout << "location: " << t << "value: " << t -> data << " next: " << t -> next << endl;
    t = t -> next;
}
```

```
● PS D:\Documents\GitHub\Introduction-to-Data-Structure\code> ./linkedlist.exe
location: 0x10f1be0value: 1 next: 0x10f1bf0
location: 0x10f1bf0value: 2 next: 0x10f1c00
location: 0x10f1c00value: 3 next: 0
❖ PS D:\Documents\GitHub\Introduction-to-Data-Structure\code> □
```



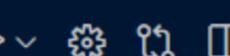
EXPLORER

...

array.cpp U

linkedlist.cpp U

full_linkedlist.cpp U X



INTRODUCTION-TO-DATA...
>.vscode
> asyncline
code
array.cpp U
array.exe U
full_linkedlist.cpp U
full_linkedlist.exe U
linkedlist.cpp U
linkedlist.exe U
pdfs
thumbnails
tutorial1
videos
.gitattributes
index.html

```
code > full_linkedlist.cpp > LinkedList > print()  
16 struct LinkedList{  
51     void print(){  
59 }  
60  
61 int main (){  
62     LinkedList l;  
63  
64     l.add(1);  
65     l.add(2);  
66     l.add(3);  
67     l.add(4);  
68  
69     l.print();  
70  
71     l.remove();  
72     l.print();  
73 }
```



PROBLEMS DEBUG CONSOLE OUTPUT PORTS SERIAL MONITOR COMMENTS TERMINAL

powershell - code + ⌂ ⌂ ⌂ ⌂

PS D:\Documents\GitHub\Introduction-to-Data-Structure\code> ./full_linkedlist.exe

1 -> 2 -> 3 -> 4 -> NULL

1 -> 2 -> 3 -> NULL

PS D:\Documents\GitHub\Introduction-to-Data-Structure\code>

OUTLINE
ASYNCLINE
VS CODE PETS

Doubly Linked List

```
struct Node{  
    int data;  
    Node* next;  
    Node* prv;  
};
```

Next Video

Stacks and Queues