

Thaneesha Fender

B1 batch

Roll no.7

Sec. D

## LINKEDLIST

```
public class LinkedListExample {
    static class Node {
        int data;
        Node next;

        Node(int data) {
            this.data = data;
            this.next = null;
        }
    }

    public static void main(String[] args) {

        Node head = new Node(5);
        Node second = new Node(10);
        Node third = new Node(15);
        Node fourth = new Node(25);

        head.next = second;
        second.next = third;
        third.next = fourth;

        Node current = head;
        while (current != null) {
            System.out.print(current.data + " -> ");
            current = current.next;
        }
        System.out.println("null");
    }
}
```


```
LinkedListExmpl...
8
9 public class LinkedListExample {
10 static class Node {
11 int data;
12 Node next;
13
14 Node(int data) {
15 this.data = data;
16 this.next = null;
17 }
18 }
19
20 public static void main(String[] args) {
21
22 Node head = new Node(5);
23 Node second = new Node(10);
24 Node third = new Node(15);
25 Node fourth = new Node(25);
26
27
28 head.next = second;
29 second.next = third;
30 third.next = fourth;
31
32
33 Node current = head;
34 while (current != null) {
35 System.out.print(current.data + " -> ");
36 current = current.next;
37 }
38 System.out.println("null");
39 }
40 }
```

input

```
5 -> 10 -> 15 -> 25 -> null

...Program finished with exit code 0
Press ENTER to exit console.
```

DB Tutorial

Save energy. Save money.  LEARN MORE >

16:08

## ARRAYLIST

```
import java.util.ArrayList;
import java.util.Scanner;
import java.time.LocalDateTime;

public class Main {
    public static void main(String[] args) {
        long startTime = System.nanoTime();
        System.out.println("Start time: " + LocalDateTime.now());

        ArrayList<String> cars = new ArrayList<>();
        cars.add("NANO");
        cars.add("RIKSHAW");
        cars.add("CYCLE");

        ArrayList<String> cc = new ArrayList<>();
        cc.add("1200");
        cc.add("1000");
        cc.add("0");

        ArrayList<String> tyres = new ArrayList<>();
        tyres.add("4");
        tyres.add("3");
        tyres.add("2");

        Scanner scanner = new Scanner(System.in);
        System.out.println("Which car?");
        String name = scanner.nextLine().toUpperCase();

        ifElse(name, cars, cc, tyres);

        long endTime = System.nanoTime();
        long elapsedTime = endTime - startTime;
        System.out.println("Elapsed time: " + elapsedTime + " nanoseconds");

        scanner.close();
    }

    static void ifElse(String name, ArrayList<String> cars, ArrayList<String> cc,
        ArrayList<String> tyres) {
        int index = cars.indexOf(name);
```

```
    if (index != -1) {  
        System.out.println("You selected: " + name);  
        System.out.println("CCs: " + cc.get(index));  
        System.out.println("Tyres: " + tyres.get(index));  
    } else {  
        System.out.println("Car not found.");  
    }  
}  
}
```



The screenshot shows a Java IDE's console window. The title bar of the console is labeled 'input'. The console output is as follows:

```
CCs: 0  
Tyres: 2  
Elapsed time: 7377245133 nanoseconds  
...Program finished with exit code 0  
Press ENTER to exit console.
```

The IDE interface includes a toolbar with icons for running, debugging, and other actions. A blue vertical bar on the left side of the console window contains the text 'at Us'.