The background is a dark blue gradient with faint, light blue circular patterns and a speaker icon. The speaker icon is located to the left of the main title. The circular patterns consist of concentric circles with arrows indicating a clockwise direction. Some of the circles have numerical labels like 160, 170, 180, 190, 200, 210, 230, 240, 250, and 260.

# TOPIC 3: LINKED LIST

CSC248 – FUNDAMENTALS OF DATA STRUCTURES

PREPARED BY: NYCJ@FSKM, UITM CAWANGAN PERLIS  
KAMPUS ARAU

# CONTENTS



Basic linked list concept



Concept in variation of  
linked list



Implementation and  
application of linked list




# BASIC LINKED LIST CONCEPT



PREPARED BY: NYCJ@FSKM, UTM CAWANGAN PERLIS KAMPUS ARAU



# ARRAYLIST VS. LINKEDLIST

 Similarity  
between  
ArrayList class &  
LinkedList class

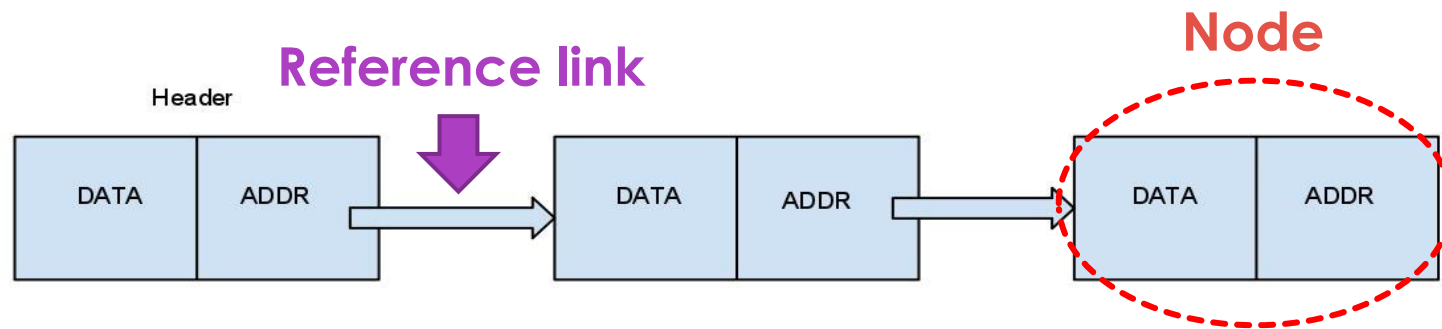
- Both classes implements List interface

 Differences between  
ArrayList class &  
LinkedList class

- LinkedList class lacks the random access feature of the ArrayList class
- LinkedList class allows constant time insertions & deletions



# INTRODUCTION TO LINKEDLIST



## Definition:

Linked list is a linear collection of objects (called nodes) connected by the reference links

## Concept:

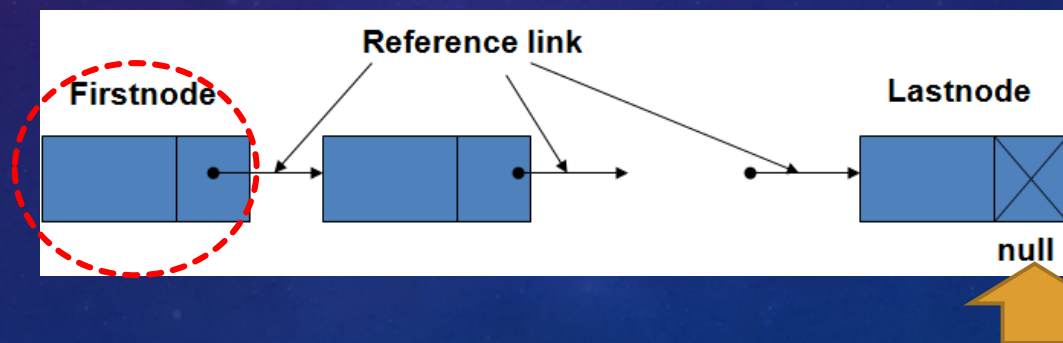
Linked list stores elements anywhere in the memory & these elements are linked by a reference/pointer

# INTRODUCTION TO LINKEDLIST



How to access a linked list?

- Program accesses a linked list via a reference to the first node in list
- Programs accesses a subsequent nodes via the reference link stored in the previous node



**Reference in the last node of a list is set to null**



# ADVANTAGES OF LINKEDLIST



Dynamic data  
structure

Simple to maintain  
elements in sorted  
order

# CONCEPT IN VARIATION OF LINKED LIST



PREPARED BY: NYCJ@FSKM, UTM CAWANGAN PERLIS KAMPUS ARAU



# TYPES OF LINKEDLIST



1

Singly linked  
list

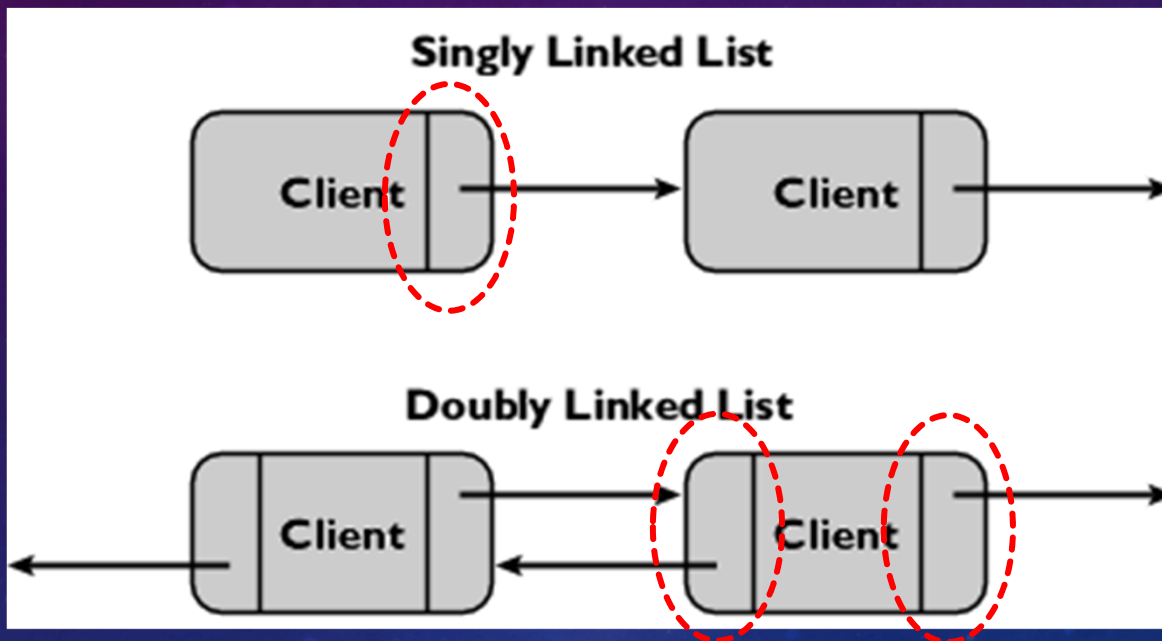
2

Doubly linked  
list

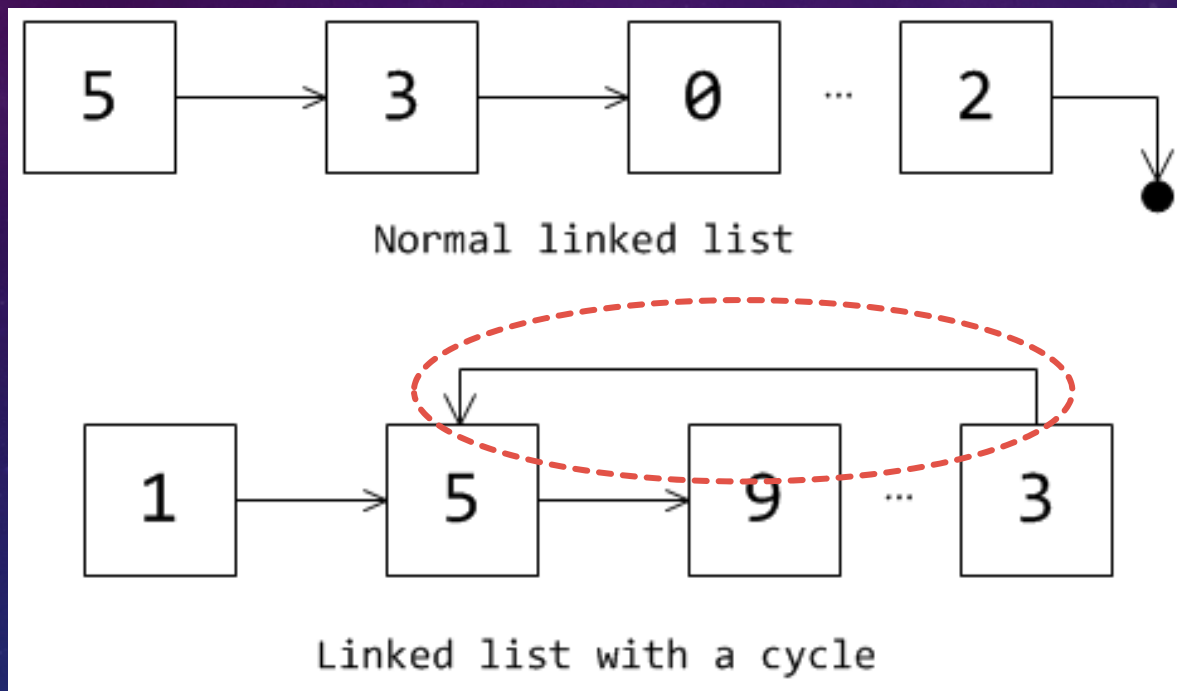
3

Circular linked  
list

# TYPES OF LINKEDLIST



# TYPES OF LINKEDLIST





# BUILT-IN METHODS FOR CLASS LINKEDLIST



- `public LinkedList()`
- `public LinkedList(Collection c)`
- `public boolean add(int index, Object element)`
- `public void addAll(Collection c)`
- `public boolean addAll(int index, Collection c)`
- `public boolean addFirst(Object element)`
- `public boolean addLast(Object element)`
- `public void clear()`
- `public boolean contains(Object element)`
- `public boolean containsAll(Collection c)`
- `public boolean equals(Object element)`
- `public Object get(int index)`
- `public Object getFirst(int index)`
- `public Object getLast(int index)`

# BUILT-IN METHODS FOR CLASS LINKEDLIST

- `public int indexOf(Object element)`
- `public boolean isEmpty()`
- `public int lastIndexOf(Object element)`
- `public boolean remove(Object element)`
- `public Object remove(int index)`
- `public boolean removeAll(Collection c)`
- `public Object removeFirst()`
- `public Object removeLast()`
- `public boolean retainAll(Collection c)`
- `public Object set(int index, Object element)`
- `public int size()`
- `public String toString()`





# EXERCISE

PREPARED BY: NYCJ@FSKM, UTM CAWANGAN PERLIS KAMPUS ARAU



## QUESTION #1



Choose the advantage of Linked List data structure.

- It is easy to maintain because the size is fixed at runtime
- The size of linked list must be given at compile.
- The portion of each node can be accessed through the index.
- It is a dynamic data structure where the size can be increased or decreased as necessary.



## QUESTION #2

The main difference between a sequential list & a linked list is:

- where data could be inserted into the list
- where data could be removed from the list
- the way data is organized in the list
- all of the above



## QUESTION #3



Which of the following is the characteristic of a circular linked list?

The null pointer at the end of the list replaced with a pointer to the beginning of the list.

Its node has 2 links; to the next & previous nodes.

It has an empty header node.

It allows data about circles & links to be stored.





## QUESTION #4



When compare to a linked list, which of the following is a DISADVANTAGE of a sequential list?

Inserting a data at the beginning of the list

Displaying a data from the list given its location

Accessing a data in the list given its location

Searching for a data in the list



# IMPLEMENTATION AND APPLICATION OF LINKED LIST



PREPARED BY: NYCJ@FSKM, UTM CAWANGAN PERLIS KAMPUS ARAU

# IMPLEMENTATION OF LINKEDLIST



Java provides class  
LinkedList for  
implementing &  
manipulating linked list  
in Java package  
(java.util.\*)

LinkedList class in Java  
uses a doubly linked list  
implementation

Programmer can also  
create self-defined  
linked list class to  
represent the list node  
& linked list structure



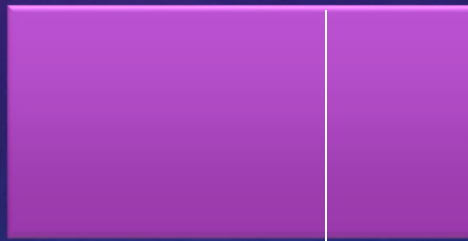
# SELF-DEFINED LINKED LIST (NODE)



## next

- Reference link
- Store reference to next node (Node type)

Node structure



ListNode object



## data

- data type (primitive or ADT)

# CLASS DEFINITION FOR NODE



---

**Class: Node**

Fields/data: data, next

Methods: Node (Object obj) //default constructor

---

```
public class Node
{
    Object data;
    Node next; //the pointer points to the next node

    Node (Object obj) //constructor to initialize the first node
    {    data = obj; } //just initialize the data, not the reference link
}
```

# SELF-DEFINED CLASS LINKED LIST



---

## **Class: LinkedList**

Fields/data: first, last, current

Methods: `LinkedList ()` //default constructor  
`boolean isEmpty()` //check whether the list is empty  
`void insertAtFront(object)` //insert at the front of the list  
`void insertAtBack(object)` //insert at the end of the list  
`Object removeFromFront()` //delete element from front  
`Object removeFromBack()` //delete element from back  
`Object getFirst()` //get the first node  
`Object getNext()` //get the reference for next node

---







**THANK YOU**

PREPARED BY: NYCJ@FSKM, UTM CAWANGAN PERLIS KAMPUS ARAU