**Day 6: 23-10-2025: Java – Collection Framework**

**Collection Framework** : Collection framework provide lot of pre defined classes and interfaces which help to store collection of object or element or data of any types like int, float, char, String or user defined objects.

It provided lot of pre defined methods which help to add, remove, search, iterate very easily.

**Collection Framework hierarchy**

**Iterable** -🡪 interface 🡪 lang package

**Collection** -🡪 Interface -🡪 util package

**Set List Queue Map**

All 4 are interfaces Set, List, Queue internally extends Collection but Map doesn’t.

**Set:** Set is an interface which is use to store collection of elements of any type. Set doesn’t allow duplicate value. Set doesn’t provide index concept. Under set few classes when we store it maintain the order or unorder or sorted by default ascending order.

**Set classes :** These classes directly or indirectly implements Set interface.

**HashSet :** HashSet super class. it doesn’t maintain the order ie UnOrder.

**LinkedHashSet:** LinkedHashSet sub class which internally extends HashSet. It maintain the order.

**TreeSet :** TreeSet internally implements SortedSet interface and that interface extends Set interface. TreeSet sorted element by default asc order.

**List**: List is an interface which allow to store collection of elements of any type. List allow index. List maintain the order. List allow duplicate.

List classes : These below classes directly or indirectly implements List interface

**ArrayList**

**ArrayList Vs Normal Array of any type**

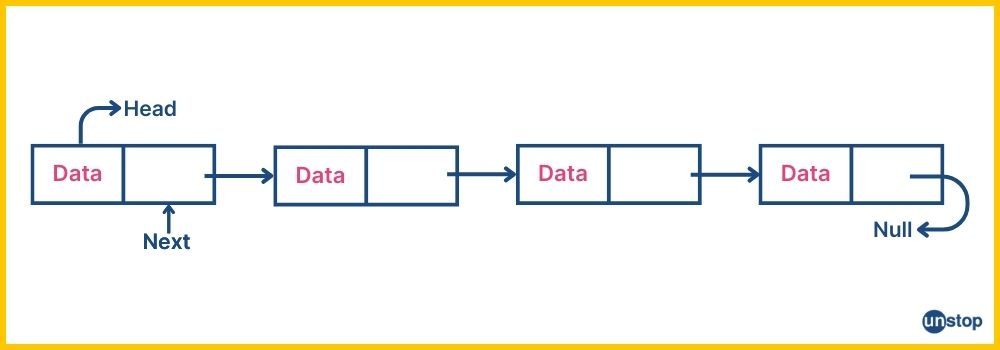
1. Normal array is known as fixed in memory. ArrayList is known as dynamic in memory.
2. Normal array allow to store homogeneous elements. ArrayList allow to store homogeneous as well as heterogeneous elements.
3. Normal Array doesn’t provide any method to store, delete, search etc method. ArrayList provided lot of pre defined methods.

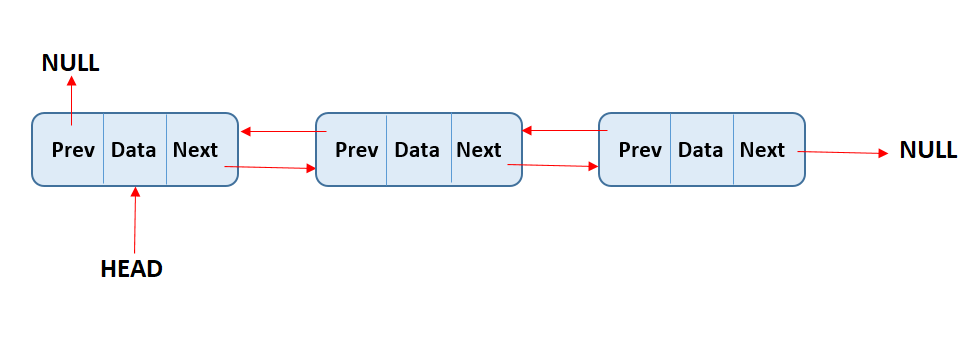
**LinkedList :** LinkedList is a type of data structure which internally use node concept to store the data.

Types of linked list

1. Singular linked list
2. Double linked list
3. Singular circular linked list
4. Double circular linked list

It hold data of any type as well as reference of node.





In Java

If we create LinkedList ll = new LinkedList()

By default it consider as double linked list

**Vector :** Vector is known as legacy class. by default all methods part of Vector class are synchronized. Performance wise slow but thread safe.

**Stack :** Stack is a type of data structure which support features as First In Last Out or Last In First Out.

Stack Operation

Push 🡪 to add the data

Pop 🡪 remove the top most data

Peek 🡪 check top most data present in stack

Stack over flow

Stack is under flow

isEmpty()🡪

size() 🡪

**Queue:** Queue is one the data structure. Which support features as First In First Out.

**Queue classes:** These below classes directly or indirectly implements Queue interface.

**PriorityQueue**

**LinkedList**

**ArrayDeque**

**Map:** it allow to store key-value pairs. Key is unique and value can be duplicate.

**Map classes:** These below classes directly or indirectly implements Map interface.

**HashMap**

**LinkedHashMap**

**TreeMap**

**Hashtable**

**Wrapper classes**

Wrapper classes primitive data types

Byte byte

Short short

Integer int

Long long

Float float

Double double

Character char

Boolean boolean

Use of wrapper classes

1. Type casting : converting to object to primitive and vice-versa.
2. It provided lot of method which can apply on those values.

Whenever we display any user defined class object reference using println method. it internally call toString() of object class. that class return String message as [packagename.className@code](mailto:packagename.className@code).

If we need meaning full object then inside user defined class you need to override toString method and return meaning full string message base upon you requirement.

Retrieve the data from collection framework one by one

1. For each loop or enhanced loop
2. Iterator
3. ListIterator
4. Enumeration 🡪 Legacy Iterator

Iterator, ListIterator and Enumeration are interfaces.

**Collection Framework with Generics**

Syntax

CollectionClass<Type> objectname = new ClassName<Type>();

Type can be All wrapper classes and user defined objects.

List<Integer> ll1 = new ArrayList<Integer>();

List<Float> ll2 = new ArrayList<Float>();

List<String> ll3 = new ArrayList<String>();

List<Object> ll4 = new ArrayList<Object>();

Collection Framework with user defined objects like Product, Customer, Employee, Account, Orders etc.

Product Management System using collection framework

Product Java Bean class : pid, pname, price etc.

Setter and getter methods

Constructor empty as well as parameter

toString() method

ProductService class which contains business methods.

App class

Collection framework provided two utilities classes.

1. Arrays
2. Collections

These two classes contains set of static method which help to do searching and sorting concept on primitive array as well as List of data of any types.

**Arrays** class provided set of methods which takes primitive array of any types as parameter and do some operation on those values like sorting, searching, comparing etc.

By default all wrapper classes as well as string class internally implements Comparable interface. This interface is a part of lang package. which provide compareTo method which help to do sorting.

**Comparator**

**Comparable Comparator**

Part of lang package part of util package

compareTo method compare method

current object and passing object two objects

same class need to different class

implements and provide the implement the interface

logic provide the logic

by default all wrapper classes, user defined class doesn’t implements Comparator interface.