

Self learning

Lesson 1

Lesson 2

Lesson 3 : basic java

Lesson 4 : Java OOPS Concept

Lesson 6 : packages and multi threading

Lesson 7 : collection framework

List interface

Stack

ArrayList

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

Node are divided into two types base upon type of linked list

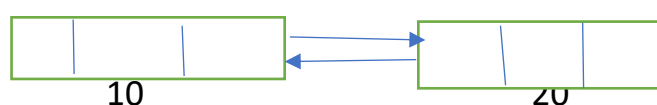
1. Singular linked list
2. Double linked list
3. Circular linked list



Value	ref of another node	value	ref of another node
-------	---------------------	-------	---------------------

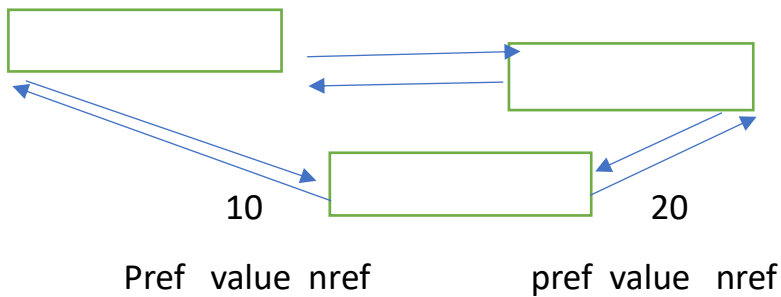
10

20



Pref value nref

```
pref value nref
```



By default java internally use double linked list concept.

Retrieve purpose array list is good option.

Inter and delete operation linkedlist list is good options.

Map : it allow to store information in the form of key-value pairs.

Key can be unique and value may be duplicate.

HashMap : display unordered

LinkedHashMap : maintain the order

TreeMap : ascending order as a key. In TreeMap we need key must be same type.

```
Set ss = new HashSet();
```

```
ss.add(10);
```

```
ss.add("Ravi");
```

```
ss.add(10.20);
```

before storing any value in any collection classes. They are consider as primitive value but once we stored it automatically convert to **Object**.

Auto boxing : converting primitive to object.

If we want to retrieve the value from any one of the collection we can take the help of

1. For each loop
2. Iterator : interface
3. ListIterator : interface

Iterator : it is an interface which contains set of method which help to retrieve the elements one by one from set and list not map.

Internal logic of for each loop base upon iterator.

In Set we can use for each as well as iterator

In list we can use for each, iterator and listiterator.

Iterator : only forward direction

ListIterator : forward as well as backward direction.

Map we can't use for each loop, Iterator as well as ListIterator.

Collection framework with Generics

```
CollectClass/Interface<Type> objectName = new ClassName<Type>();
```

Type can be Integer, Float, Character, Double, String as well as user defined class object.

```
List<Integer> ll = new ArrayList<Integer>();
```

Collection Framework with user defined class object ie JavaBean.