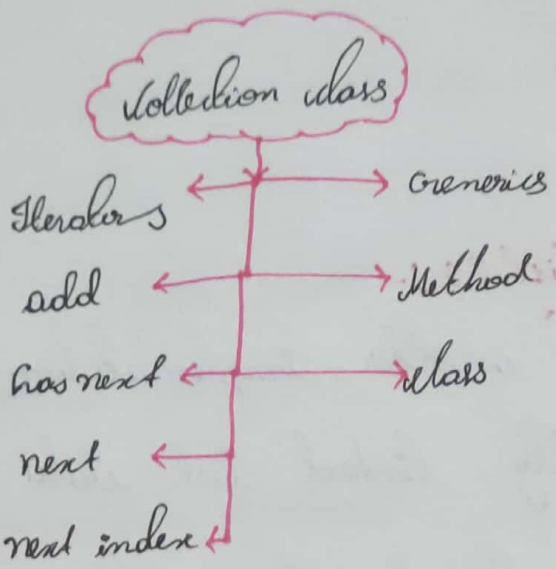
CSA0992 Java Program

G. Krishna Lal
192011130
CSE

Assignment - 5

Collection classes :-



1) Array list :-

- It is a dynamic array based implements of the list interface.

```
import java.util.Arraylist;
Public static void main (Strings [] args) {
Arraylist < string > names = new Arraylist <> ();
    names.add ("Alice");
    names.add ("Bob");
    names.add ("charlie");
    System.out.Println ("Names:" + names);
        names.remove ("Bob");
```

```
        System.out.println("Names after remove bob:" + names);
    }
}
```

2) LinkedList():-

- It is another implementation of list interface, based on doubly-linked list data structure.
- It provides efficient insertion and deletion of elements at both ends of the list.
- Here's a simple program that demonstrates it.

```
import java.util.LinkedList;
Public class LinkedList {
    Public static void main (Strings [] args) {
LinkedList<Integer> number = new LinkedList<>();
    number.add(1);
    number.add(2);
    numbr.add(3);
        System.out.println("Numbrs after remove:"
                                + numbrs);
    }
}
```

3) Map Interface :-

* HashMap :-

→ It is an implementation of the Map interface that stores key values Pairs.

→ It uses a hash table to Provide fast acess to value based on their keys.

```
impat Java.util.hashmap;
  Publie dass hash {
    Publie static void main (String [] args) {
    Hash < String , integer > scores = new hashmap <>();
    Hores. Put ("Alice, 95);
    Sores. Put ("Bob", 87);
    Hores. Put ("charlie", 91);
    System. out. Println (" Bob's Score: " + Scores. get ("Bob"));
      Hores - remoues ("charle");
      tyetem. Out. Println ("Hash map after remove:" + Scores);
  }
}
```

+ Linked Hash Map :-

- It is another implementation of map interface.
- It maintains the order of key value pairs based on the insertion order.

```
import java.util.LinkedHashMap;
Public class Linked {
    Public static void main (String[] args) {
    LinkedHashMap < String, Integer > ages = new Linked<>();

    ages.Put ("Alice", 25);
    ages.Put ("Bob", 30);
    ages.Put ("Charlie", 22);

      System.out.Println ("Ages: " + ages);

    ages.remove("Bob");
    System.out.Println("LinkedHashMap after removing Bob:" +
                                        ages);

    }
}
```