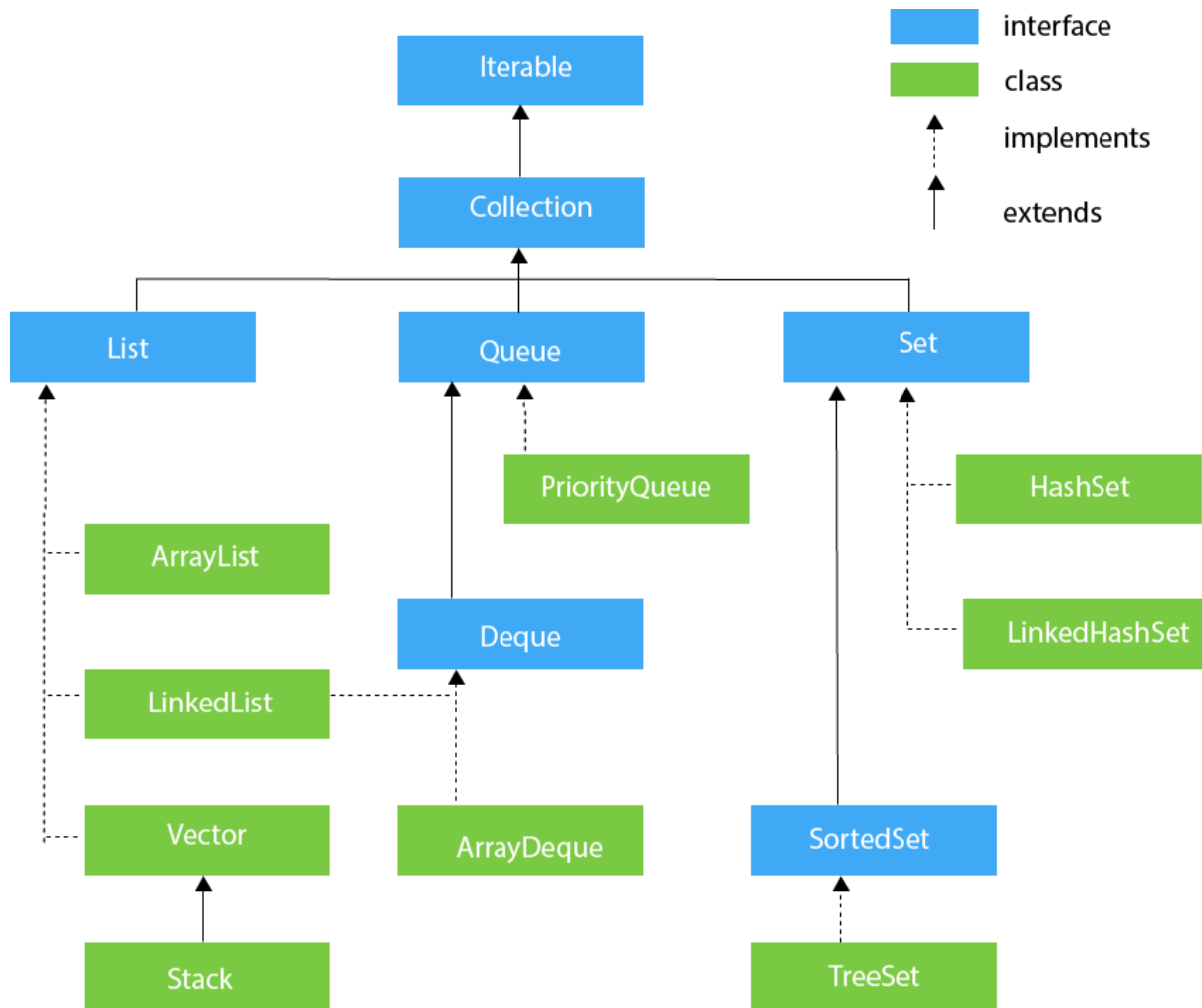


SESSION 11 PROGRAMS(Collection Framework)

Hierarchy of Collection Framework



Program 1 ArrayList

```
1. import java.util.*;
2. class TestJavaCollection1{
3.     public static void main(String args[]){
4.         ArrayList<String> list=new ArrayList<String>();//Creating arraylist
5.         list.add("Ravi");//Adding object in arraylist
6.         list.add("Vijay");
7.         list.add("Ravi");
8.         list.add("Ajay");
9.         //Traversing list through Iterator
10.        Iterator itr=list.iterator();
11.        while(itr.hasNext()){
12.            System.out.println(itr.next());
13.        }
14.    }
15. }
```

Program2 Vector

```
1. import java.util.*;
2. public class TestJavaCollection3{
3.     public static void main(String args[]){
4.         Vector<String> v=new Vector<String>();
5.         v.add("Ayush");
6.         v.add("Amit");
7.         v.add("Ashish");
8.         v.add("Garima");
9.         Iterator<String> itr=v.iterator();
10.        while(itr.hasNext()){
11.            System.out.println(itr.next());
12.        }
13.    }
14. }
```

Program3 HashSet

```
import java.util.*;
1. public class TestJavaCollection7{
2.     public static void main(String args[]){
3.         //Creating HashSet and adding elements
4.         HashSet<String> set=new HashSet<String>();
5.         set.add("Ravi");
6.         set.add("Vijay");
7.         set.add("Ravi");
8.         set.add("Ajay");
9.         //Traversing elements
10.        Iterator<String> itr=set.iterator();
11.        while(itr.hasNext()){
12.            System.out.println(itr.next());
13.        }
14.    }
15. }
```

Program4 Java Map Example

```
1. import java.util.*;
2. class MapExample2{
3.     public static void main(String args[]){
4.         Map<Integer,String> map=new HashMap<Integer,String>();
5.         map.put(100,"Amit");
6.         map.put(101,"Vijay");
7.         map.put(102,"Rahul");
8.         //Elements can traverse in any order
9.         for(Map.Entry m:map.entrySet()){
10.            System.out.println(m.getKey()+" "+m.getValue());
11.        }
12.    }
13.}
```

Core Java Training

ArrayList	LinkedList
It uses a dynamic array as it's internal implementation.	It uses doubly linked list as it's internal implementation.
It is better in get and set operations.	It is better in adding and removing operations.

List	Set	Map
Allow duplicate elements.	Does not allow duplicate elements.	Does not allow duplicate key but values can be duplicate.
Allow multiple null values.	Allow single null value.	Allow single null as key and multiple null as values.
Maintains insertions order.	Set represents an unordered collection but some of its implementation classes maintains some order. LinkedHashSet maintains order, TreeSet maintains ascending order.	Like Set, Map also represents an unordered collection. Again same like Set, some of its implementation classes maintains some order. TreeMap maintains ascending order of keys.

SESSION 11 ASSIGNMENTS

1. Write a Java program to create a new array list, add some colors (string) and print out the collection.
2. Write a Java program to iterate through all elements in a hash set, add some colors (string) and print out the collection
3. Write a Java program to get the value of a specified key in a hash map. Store ID, Name, Address as keys & their corresponding values