Q1. Write a program to traverse (or iterate) ArrayList?

```
import java.util.*;
public class ArrayListLoopExample {
    public static void main(String args[]) {

        // initialize ArrayList
        ArrayList<Integer> al = new ArrayList<Integer>();

        // add elements to ArrayList object
        al.add(3);
        al.add(17);
        al.add(6);
        al.add(9);
        al.add(7);
        System.out.println("Using Advanced For Loop");
        // printing ArrayList
        for (Integer num : al) {
            System.out.println(num);
        }
    }
}
```

Output on Console:

```
Using Advanced For Loop
3
17
6
9
7
```

Q2 Write a program to convert List to Array

```
import java.util.*;
```

```
public class ConvertArrayListToArray {

  public static void main(String args[]) {

   // Creating and initializing ArrayList
   ArrayList<String> fruits = new ArrayList<>();
   fruits.add("Apple");
   fruits.add("Banana");
   fruits.add("Mango");
   fruits.add("Pear");

   // ArrayList to String array conversion
   String[] str = new String[fruits.size()];
   for(int i=0; i < fruits.size(); i++) {
     str[i] = fruits.get(i);
   }

   // Print elements using for-each loop
   for(String s : str) {
    System.out.println(s);
   }
  }
}
```

Output on Console:

```
Apple
Banana
Mango
Pear
```

Q3 Write a program to traverse (or iterate) HashSet?

```
import java.util.*;

 public class HashSetIteratorExample {
    public static void main(String args[]) {

    // Declaring a HashSet
    HashSet<String> hashset = new HashSet<String>();
    // Add elements to HashSet
    hashset.add("Pear");
    hashset.add("Apple");
    hashset.add("Orange");
    hashset.add("Papaya");
    hashset.add("Banana");
    // Get iterator
    Iterator<String> it = hashset.iterator();
    // Show HashSet elements
    System.out.println("HashSet contains: ");
    while(it.hasNext()) {
      System.out.println(it.next());
    }
  }
}
```

Output on Console:

```
HashSet contains:
Apple
Pear
Papaya
Orange
Banana
```

Q4 Given an element write a program to check if element (value) exists in ArrayList?

```
import java.util.*;
public class ArrayListContainsExample {
    public static void main(String args[]) {
        // initialize ArrayList
        ArrayList<Integer> al = new ArrayList<Integer>();
        // add elements to ArrayList object
        al.add(3);
        al.add(17);
        al.add(6);
        al.add(9);
        al.add(7);
        // check if ArrayList contains element
        if (al.contains(7)) {
            System.out.println("7 was found in the list");
        } else {
            System.out.println("7 was not found in the list");
        }
    }
}
```

Output on Console:

```
7 was found in the list
```

Q5 Given an element write a program to check if element exists in HashSet?

```
import java.util.*;
public class HashSetContainsExample {
    public static void main(String args[]) {
        // initialize HashSet
        HashSet<Integer> hs = new HashSet<Integer>();
        // add elements to HashSet object
        hs.add(3);
        hs.add(17);
        hs.add(6);
        hs.add(9);
        hs.add(7);
        // check if HashSet contains element
        if (hs.contains(7)) {
            System.out.println("7 was found in the list");
        } else {
            System.out.println("7 was not found in the list");
        }
    }
}
```

Output on Console:

```
7 was found in the list
```

Q6 Write a program to initialize a HashMap in java ?

```
   // initialize HashMap
        HashMap<String,String> hashmap = new HashMap<String,String>();
```

Q7 Write a program to initialize an ArrayList in java?

```
// initialize ArrayList
        ArrayList<Integer> al = new ArrayList<Integer>();
```

Q8 Write a program to convert Array to List?

Method 1 : Using Arrays.asList() method

Syntax :

```
ArrayList<String> list = new ArrayList<>(Arrays.asList(arrayname));
```

```
import java.util.*;

public class ConvertArrayToArrayList {
    public static void main(String args[]) {
      // Declaring and initializing Array
      String[] cities={"Boston", "Dallas", "New York", "Chicago"};
      //Converting Array to ArrayList using Arrays.asList()
      ArrayList<String> list= new ArrayList<>(Arrays.asList(cities));

      // Add more elements to the converted list
      list.add("San Francisco");
      list.add("San jose");

      // Print arraylist elements using for-each loop
      for(String s : list) {
        System.out.println(s);
      }
    }
}
```

Output on Console:

```
Boston
Dallas
New York
Chicago
San Francisco
San jose
```

Method 2 : Using Collections.addAll() method

Syntax:

```
Collections.addAll(arraylist, array);
```

```
import java.util.*;

public class ConvertArrayToArrayList2 {
    public static void main(String args[]) {
```

```
   // Creating and initializing Array
   String[] strArray = {"AAA", "BBB", "CCC", "DDD"};

   // Declaring ArrayList
   ArrayList<String> al = new ArrayList<>();

   //Converting Array to ArrayList using addAll() method
   Collections.addAll(al, strArray);

   // Add more elements to the converted list
   al.add("YYY");
   al.add("ZZZ");

   // Displaying arraylist elements using for-each loop
   for(String s : al) {
    System.out.println(s);
   }
  }
}
```

Output on Console:

```
AAA
BBB
CCC
DDD
YYY
ZZZ
```

Method 3 : Using add() method

```
import java.util.*;

public class ConvertArrayToArrayList3 {
  public static void main(String args[]) {
    // Declaring and instantiating ArrayList in one step
    ArrayList<String> al = new ArrayList();

    // Given initialized array
    String[] strArray = {"Cocacola", "Pepsi", "Fanta", "Dr Pepper"};

    //Converting Array to ArrayList manually
    for (int i=0; i < strArray.length ; i++) {
      // Adding every element of array to the ArrayList
      al.add(strArray[i]);
    }

    // Showing arraylist elements using for-each loop
    for(String str1 : al) {
     System.out.println(str1);
    }
  }
}
```

Output on Console:

```
Cocacola
Pepsi
Fanta
Dr Pepper
```

Q9 Write a program to find the length of the ArrayList?

1. Java Program to find Length/Size of Integer ArrayList

```
import java.util.*;
import java.io.*;

/* Write a program to determine the size/length of the ArrayList*/
public class ArrayListSize
{
    public static void main (String[] args)
    {
        // Create an Integer ArrayList Object
        ArrayList<Integer> arrlist=new ArrayList<Integer>();

        // Print initial size of ArrayList
        System.out.println("Size before adding elements: "+arrlist.size());

        // Adding elements to ArrayList Object
        arrlist.add(11);
        arrlist.add(3);
        arrlist.add(5);
        arrlist.add(4);
        arrlist.add(9);

        /* Print size of ArrayList
           after adding elements */
        System.out.println("Size after adding elements: "+arrlist.size());

        // Removing elements from ArrayList
        arrlist.remove(1);
        arrlist.remove(2);

        /* Print size of ArrayList
           after removing elements */
        System.out.println("Size after removing elements: "+arrlist.size());

        // Print ArrayList
        System.out.println("Resulting ArrayList: ");
        for(int num: arrlist){
            System.out.println(num);
        }
    }
```

Output on Console:

```
Size after removing elements: 3
Resulting ArrayList:
11
5
9
```

2. Java Program to find Length/Size of String ArrayList

```
import java.util.*;
import java.io.*;

/* Program to find size of ArrayList in Java */
public class ArrayListSize
{
      public static void main (String[] args)
  {
     System.out.println("Java Program to find the size of ArrayList");

     // Create an String ArrayList Object
     ArrayList<String> listOfCities = new ArrayList<>();

     int size = listOfCities.size();
     // Print initial size of ArrayList
     System.out.println("size of ArrayList after creation: " + size);

     // Adding elements to ArrayList Object
     listOfCities.add("California");
     listOfCities.add("Boston");
     listOfCities.add("New York");

     size = listOfCities.size();
     /* Print size of ArrayList
       after adding elements */
     System.out.println("size of ArrayList after adding elements: " + size);

     // clear() method removes all elements
```

```
        listOfCities.clear();

        size = listOfCities.size();
        System.out.println("size of ArrayList after clearing elements: " + size);
    }
}
```

Output on Console:

```
Java Program to find the size of ArrayList
size of ArrayList after creation: 0
size of ArrayList after adding elements: 3
size of ArrayList after clearing elements: 0
```

Q10 Write a program to add elements to the HashMap given the key and value data type is String?

```
// Declaring a HashMap of String keys and String values
   HashMap<String, String> hashmap = new HashMap<String, String>();
   // Adding key-value pairs to HashMap
   hashmap.put("1", "Value1");
   hashmap.put("2", "Value2");
   hashmap.put("3", "Value3");
   hashmap.put("4", "Value4");
   hashmap.put("5", "Value5");
```

Q11 Write a program to initialize a HashSet in java?

```
// initialize HashSet
      HashSet<Integer> al = new HashSet<Integer>();
```

Q12 Write a program to add elements to ArrayList ?

```
ArrayList<Integer> al = new ArrayList<Integer>();
        // add elements to ArrayList object
        al.add(3);
        al.add(17);
        al.add(6);
        al.add(9);
        al.add(7);
```

Q13 Write a program to add elements to HashSet?

```
import java.util.*;
public class HashSetAddExample {
    public static void main(String args[]) {
        // initialize HashSet
        HashSet<Integer> hs = new HashSet<Integer>();
        // add elements to HashSet object
        hs.add(3);
        hs.add(17);
        hs.add(6);
        hs.add(9);
        hs.add(7);
        System.out.println("Using Advanced For Loop");
        // printing HashSet
        for (Integer num : hs) {
            System.out.println(num);
        }
    }
}
```

Q14 Write a program to get size of HashMap?

```
import java.util.HashMap;
```

```
public class HashMapSizeExample {

     public static void main(String args[]) {

           // Creating HashMap object with Integer keys and String
values
           HashMap<Integer,String> map = new HashMap<>();

           // Adding elements to the HashMap object
           map.put(1, "CocoCola");
           map.put(2, "Pepsi");
           map.put(3, "Thums Up");
           map.put(4, "Fanta");

           // Calculating the size of the HashMap using size() method
           System.out.println(" Size of the given HashMap is: "+
map.size());
         }
}
```

Output on Console:

```
Size of the given HashMap is: 4
```

```
import java.util.HashMap;

public class HashMapSizeExample2 {

     public static void main(String args[]) {

           // Creating HashMap object with String keys and Integer
values
           HashMap<String, Integer> map2 = new HashMap<>();

           // Putting elements to the HashMap object
           map2.put("Java", 10);
           map2.put("Hungry", 20);
           map2.put("Blog", 30);

           // Finding the size of the HashMap using size() method
           System.out.println(" Size of the given HashMap is: "+
map2.size());
         }
}
```

Output on Console:

Size of the given HashMap is: 3

Q15 How to check if HashMap is empty?

1. Using isEmpty() method

```
import java.util.HashMap;

public class HashMapEmptyExample {

      public static void main(String args[]) {

            // Creating HashMap object with Integer keys and String 
values
            HashMap<Integer, String> map = new HashMap<>();

            // Checking whether HashMap is empty or not
            System.out.println("Checking Is HashMap empty?: " + 
map.isEmpty());

            // Adding elements to the HashMap object
            map.put(100, "Jack");
            map.put(200, "John");
            map.put(300, "Smith");

            // Checking again whether HashMap is empty or not
            System.out.println("Checking Is HashMap empty?: "+ 
map.isEmpty());
          }
}
```

Output on Console:

```
Checking Is HashMap empty?: true
Checking Is HashMap empty?: false
```

2. Using size() method

```
import java.util.HashMap;

public class HashMapEmptyExample2 {

      public static void main(String args[]) {

            // Creating HashMap object with String keys and String
values
            HashMap<String, String> map = new HashMap<>();

            // Checking whether HashMap is empty or not using size()
method
            System.out.println("Checking Is HashMap empty using size()
method?: " + (map.size()==0));

            // Putting elements to the HashMap object
            map.put("100", "Java");
            map.put("1000", "Python");
            map.put("10000", "Javascript");

            // Checking again whether HashMap is empty or not using
size() method
            System.out.println("Checking Is HashMap empty using size()
method?: "+ (map.size()==0));
          }
}
```

Output on Console:

```
Checking Is HashMap empty using size() method?: true
Checking Is HashMap empty using size() method?: false
```

Q16 Write a program to iterate the HashMap ?

Iterating or looping Map Using keySet() and foreach loop

```
import java.util.HashMap;

public class HashMapLoopExample {

        public static void main(String args[]) {

                // Creating a HashMap of String keys and String values
                HashMap<String, String> hashmap = new HashMap<String,
String>();
                hashmap.put("Key1", "Value1");
                hashmap.put("Key2", "Value2");
                System.out.println("Iterating or looping map using
foreach loop");
                // Iterating or looping using keySet() method
                for (String key : hashmap.keySet()) {
                    System.out.println("key: " + key + " value: " +
hashmap.get(key));
                }
             }
}
```

Output on Console:

```
Iterating or looping map using foreach loop
key: Key2 value: Value2
key: Key1 value: Value1
```

Q17 Write a program to sort HashMap by keys ?

```
import java.util.HashMap;
import java.util.Iterator;
import java.util.Map;
import java.util.Set;
import java.util.TreeMap;

public class HashMapSortByKeyExample {
      public static void main(String args[]) {

               // Creating a HashMap of int keys and String values
               HashMap<Integer, String> hashmap = new HashMap<Integer,
String>();

               // Adding Key and Value pairs to HashMap
               hashmap.put(22,"A");
               hashmap.put(55,"B");
               hashmap.put(33,"Z");
               hashmap.put(44,"M");
               hashmap.put(99,"I");
               hashmap.put(88,"X");

               System.out.println("Before Sorting:");
               Set set = hashmap.entrySet();
               Iterator iterator = set.iterator();
               while(iterator.hasNext()) {
                   Map.Entry pair = (Map.Entry)iterator.next();
                   System.out.print(pair.getKey() + ": ");
                   System.out.println(pair.getValue());
               }
               TreeMap<Integer, String> map = new TreeMap<Integer,
String>(hashmap);
               System.out.println("After Sorting:");
               Set set2 = map.entrySet();
               Iterator iterator2 = set2.iterator();
               while(iterator2.hasNext()) {
                   Map.Entry pair = (Map.Entry)iterator2.next();
                   System.out.print(pair.getKey() + ": ");
                   System.out.println(pair.getValue());
               }
             }
}
```

Output on Console:

```
Before Sorting:
33: Z
99: I
22: A
55: B
88: X
44: M
```

```
After Sorting:
22: A
33: Z
44: M
55: B
88: X
99: I
```

Q18 Write a program to sort ArrayList using Comparable and Comparator?

```java
import java.util.Comparator;

public class ComparatorDiscussion implements Comparator<String> {

    @Override
    public int compare(String o1, String o2) {

        return o1.compareTo(o2);
    }

}
```

```java
import java.util.TreeSet;

public class TreeSetClass {

    public static void main(String[] args) {


        TreeSet<String> jk = new TreeSet<String>(new
ComparatorDiscussion());

        jk.add("Riddhi");

        jk.add("Siddhi");

        jk.add("Vedant");

        jk.add("Badri");
```

```
           jk.add("Digu");

           System.out.println(jk);

           TreeSet<Integer> jk1 = new TreeSet<Integer>(new 
ComparatorDiscussion2());

           jk1.add(4);

           jk1.add(8);

           jk1.add(2);

           jk1.add(1);

           System.out.println(jk1);

      }
}
```

Output on Console:

```
[Badri, Digu, Riddhi, Siddhi, Vedant]
[8, 4, 2, 1]
```

Q19 Write a program to sort ArrayList in descending order?

```
import java.util.ArrayList;
import java.util.Collections;

public class ArrayListDescendingSort {

public static void main(String args[]) {

        ArrayList<String> arrList = new ArrayList();
        arrList.add("Apple");
        arrList.add("Banana");
        arrList.add("Pear");
        arrList.add("Mango");

        /*Unsorted List: ArrayList content before sorting*/
        System.out.println("ArrayList Before Sorting:");
        for(String s: arrList){
           System.out.println(s);
        }

        /* Sorting in decreasing (descending) order*/
        Collections.sort(arrList, Collections.reverseOrder());

        /* Sorted List in reverse order*/
        System.out.println("ArrayList in descending order:");
        for(String str: arrList){
           System.out.println(str);
        }
  }
}
```

Output on Console:

```
ArrayList Before Sorting:
Apple
Banana
Pear
Mango
ArrayList in descending order:
Pear
Mango
Banana
Apple
```

Q20 Write a program to add element at particular index of ArrayList?

1. Add String elements at the specified index in ArrayList

```
import java.util.ArrayList;

public class AddArrayListExample {

     public static void main(String args[]) {

            // Declaration of String ArrayList
            ArrayList<String> al = new ArrayList<String>();

            /* Simple add() method for adding element
               at the end of the ArrayList */
            al.add("California");
            al.add("Boston");
            al.add("San jose");
            al.add("New York");

            //Adding element to the 3rd position
            //3rd position = 2 index as index starts with 0
            al.add(2,"San Francisco");
            System.out.println("ArrayList after adding String San
Francisco:"+ al);

            //Addition of String element at 1st position
            al.add(0, "Texas");

            //Displaying the ArrayList
            System.out.println("ArrayList after adding String Texas:"+
al);
         }
}
```

Output on Console:

```
ArrayList after adding String San Francisco:[California, Boston, San
Francisco, San jose, New York]
ArrayList after adding String Texas:[Texas, California, Boston, San
Francisco, San jose, New York]
```

Q21 Write a program to remove element from specified index of ArrayList?

```
import java.util.ArrayList;

public class RemoveMethodExample {

      public static void main(String args[]) {
             // Creating an object of ArrayList of String Type
             ArrayList<String> list = new ArrayList<>();
             list.add("AA");
             list.add("BB");
             list.add("CC");
             list.add("DD");
             list.add("AA");
             list.add("ZZ");
             System.out.println("Given ArrayList before remove method:
");
             for(String str : list) {
                 System.out.println(str);
             }
       // Using remove(element) method, removing 1st element, size()
reduces by 1
             list.remove(0);

        // Using remove(element) method, removing 3rd element from the
remaining list
             list.remove(2);

       // Using remove(element) method, removing 4th element from the
remaining list
             list.remove(3);
             System.out.println("ArrayList after applying remove
method: ");
             for(String str2 : list) {
                 System.out.println(str2);
             }
         }
}
```

Output on Console:

```
Given ArrayList before remove method:
AA
BB
CC
```

```
DD
AA
ZZ
ArrayList after applying remove method:
BB
CC
AA
```

Q22 Write a program to convert LinkedList to ArrayList?

```
import java.util.ArrayList;
import java.util.LinkedList;
import java.util.List;

public class LinkedListToArrayList {

      public static void main(String args[]) {

               // Creating LinkedList Object
               LinkedList<String> linkedlist = new
LinkedList<String>();
               linkedlist.add("Mango");
               linkedlist.add("Banana");
               linkedlist.add("Pear");
               linkedlist.add("Apple");
               linkedlist.add("Orange");

               // Converting LinkedList to ArrayList
               List<String> list = new ArrayList(linkedlist);

               for (String s : list) {
                 System.out.println(s);
               }
             }
}
```

Output on Console:

```
Mango
Banana
Pear
```

```
Apple
Orange
```

Q23 Write a program to convert HashSet to Array?

```java
import java.util.HashSet;

public class HashSetToArray {

      public static void main(String args[]) {

            // Create a HashSet object
            HashSet<String> hashset = new HashSet<String>();

            // Adding elements to HashSet object
            hashset.add("Doctor");
            hashset.add("Engineer");
            hashset.add("Lawyer");
            hashset.add("Police");

            // Printing HashSet elements
            System.out.println("HashSet contains: "+ hashset);

            // Creating an Array of HashSet size
            String[] array = new String[hashset.size()];

            // Converting HashSet to Array using toArray() method
            hashset.toArray(array);

            // Printing Array elements
            System.out.println("Array contains: ");
            for (String str : array) {
                System.out.println(str);
            }
          }
}
```

Output on Console:

```
HashSet contains: [Engineer, Doctor, Lawyer, Police]
Array contains:
```

```
Engineer
Doctor
Lawyer
Police
```

Q24 Write a program to reverse ArrayList in java?

```java
import java.util.ArrayList;
import java.util.Collections;

public class ReverseArrayList {

      public static void main(String[] args)
         {
             //Creating an ArrayList object
             ArrayList<String> arrlist = new ArrayList<String>();

             //Adding elements to ArrayList object
             arrlist.add("Apple");
             arrlist.add("Amazon");
             arrlist.add("Facebook");
             arrlist.add("Google");
             arrlist.add("IBM");
             arrlist.add("Tesla");

             //Displaying ArrayList Before Reverse
             System.out.println("Before Reverse ArrayList:");
             System.out.println(arrlist);

             /*Reversing the list using
               Collections.reverse() method*/
             Collections.reverse(arrlist);

             //Displaying list after reverse
             System.out.println("After Reverse ArrayList:");
             System.out.println(arrlist);
         }
}
```

Output on Console:

```
Before Reverse ArrayList:
[Apple, Amazon, Facebook, Google, IBM, Tesla]
After Reverse ArrayList:
[Tesla, IBM, Google, Facebook, Amazon, Apple]
```

Q25 Write a program to iterate TreeMap in java?

```
import java.util.Iterator;
import java.util.Map;
import java.util.Set;
import java.util.TreeMap;

public class TreeMapIteratorExample {


      public static void main(String args[]) {

               // Declaring a TreeMap of String keys and String values
               TreeMap<String, String> treemap = new TreeMap<String, 
String>();

               // Add Key-Value pairs to TreeMap
               treemap.put("Key1", "Pear");
               treemap.put("Key2", "Apple");
               treemap.put("Key3", "Orange");
               treemap.put("Key4", "Papaya");
               treemap.put("Key5", "Banana");


               // Get Set of entries
               Set set = treemap.entrySet();

               // Get iterator
               Iterator it = set.iterator();

               // Show TreeMap elements
               System.out.println("TreeMap contains: ");
               while(it.hasNext()) {
                 Map.Entry pair = (Map.Entry)it.next();
                 System.out.print("Key is: "+pair.getKey() + " and ");
                 System.out.println("Value is: "+pair.getValue());
               }
             }
}
```

Output on Console:

```
TreeMap contains:
Key is: Key1 and Value is: Pear
Key is: Key2 and Value is: Apple
Key is: Key3 and Value is: Orange
Key is: Key4 and Value is: Papaya
Key is: Key5 and Value is: Banana
```