

Q4] How do you find the number of elements present in an arraylist?

Using size() method. size() method returns number of elements present in an ArrayList.

```
import java.util.*;  
class arrayListA3  
{  
    public static void main (String args [])  
    {  
        ArrayList<String> arr2 = new ArrayList<String>();  
        arr2.add ("Java");  
        arr2.add ("Programming");  
        arr2.add ("language");  
        System.out.println ("Number of elements present in arraylist is "+arr2.size());  
        System.out.println ("Elements in given arraylist ".+arr2.size());  
    }  
}
```

Assignment 4

Q.1] Explain the different ways of constructing an arraylist ?

```
import java.io.*;
import java.util.*;
class ArrayListConstructors {
    public static void main (String args[])
    {
        ArrayList<Integer> array1 = new ArrayList<Integer>(3);
        ArrayList class
        List<String> array2 = new ArrayList<>(4);
        ArrayList using list interface
        array1.add(1);
        array1.add(2);
        array1.add(3);
        array2.add("Java");
        array2.add("programming");
        System.out.println("Elements of first
arraylist:" + array1);

        System.out.println("Elements of second
arraylist :" + array2);
    }
}
```

Q.2] How do you increase the current capacity of an ArrayList?

`ensureCapacity()` method is used to increase the current capacity of an ArrayList. However, capacity of an ArrayList is automatically increased when we try to add more elements than the current capacity, `ensureCapacity()` method is used.

```
→ import java.io.*;
import java.util.*;
class arrayList A1 {
    public static void main (String args[])
    {
        ArrayList < Integer > arrlist = new ArrayList
        < Integer > (5);
        arrlist.add (10);
        arrlist.add (50);
        arrlist.add (30);
        for (integer number : arrlist) {
            System.out.println ("Number = " + number);
        }
    }
}
```

Q.3] How do you decrease the current capacity of an ArrayList to the current size?

trimToSize() method is used to trim the capacity of ArrayList to the current size of

ArrayList. Developers use this method to minimize the storage area of an ArrayList.

```
→ import java.util.*;
class ArrayListA2
{
    public static void main (String args[])
    {
        ArrayList<Integer> arr2 = new ArrayList();
        arr2.add(10);
        arr2.add(20);
        arr2.add(30);
        arr2.add(40);
        arr2.trimToSize(); // trim the ArrayList
        System.out.println ("Elements in given
                            arraylist "+arr2);
    }
}
```

Q4] How do you find the number of elements present in an arraylist?

Using size() method. size() method returns number of elements present in an ArrayList.

```
import java.util.*;  
class arrayListA3  
{  
    public static void main (String args[])  
    {  
        ArrayList<String> arr2 = new ArrayList<  
        arr2.add ("Java");  
        arr2.add ("Programming");  
        arr2.add ("language");  
        System.out.println ("Number of elements  
        present in arraylist is = "+arr2.size());  
        System.out.println ("Element in given  
        arraylist ".+arr2.size());  
    }  
}
```

5) How do you find out whether the given arraylist is empty or not?

isEmpty() method of ArrayList is used to check whether the given ArrayList is empty or not. This method returns true if an ArrayList contains no elements otherwise returns false.

```
import java.util.*;  
class ArrayListAU  
{  
    public static void main (String args [])  
    {  
        ArrayList < Integer > arr2 = new ArrayList < ( );  
        arr2.add (10);  
        arr2.add (20);  
        arr2.add (30);  
        boolean result = arr2.isEmpty ();  
        if (result = true)  
        {  
            System.out.println ("ArrayList is empty");  
        }  
        else  
        {
```

```
System.out.println ("arraylist is not empty");
System.out.println ("arraylist is not
empty");
```

```
System.out.println ("elements in the array
list are": +arr2);
```

}

}

}

Q.6] How do you check whether the given element is present in an arraylist or not?

Using contains () method of arraylist, we can examine whether the arraylist contains the given element or not. This method returns true if ArrayList has that element otherwise returns false.

```
import java.util.*;
class ArrayListAS
{
    public static void main (String args[])
    {
        ArrayList<Integer> arr2 = new ArrayList<>();
        arr2.add(10);
        arr2.add(20);
        arr2.add(30);
```

```
arr2.add(40);
System.out.println("enter the element to
Search");
Scanner s= new Scanner(System.in);
int a= s.nextInt();
if(arr2.contains(a))
{
    System.out.println("arraylist contains "+a);
}
else
{
    System.out.println("arraylist does not contain
"+a);
}
```

Q.7] How do you get the position of a particular element in an arraylist?
We can use indexOf() and lastIndexOf()
method to find out the position of a given element in an arraylist. indexOf
method returns index of last occurrence
of a specified element in an arraylist.
if element is found, they will return

```
import java.util.*;  
class arraylistAG  
{  
    public static void main (String args[])  
    {  
        ArrayList<Integer> arr2 = new ArrayList<>(5);  
        arr2.add(10);  
        arr2.add(20);  
        arr2.add(30);  
        arr2.add(40);  
        System.out.println ("enter the element to -  
                           index of it");  
        Scanner s = new Scanner (System.in);  
        int a = s.nextInt();  
        int b = arr2.indexOf(a);  
        System.out.println ("index of "+a+" is "+b)  
    }  
}
```

Q] How do you convert an arraylist to an array? The `toArray()` method returns an array containing all elements of the arraylist. This method acts as a bridge between normal arrays and collection framework in Java.

```
import java.util.*;  
public class arrayListA7  
{  
    public static void main (String args [])  
    {  
        ArrayList < Integer > Arrlist = new ArrayList  
        < Integer > ();  
        Arrlist.add (15);  
        Arrlist.add (25);  
        Arrlist.add (35);  
        Arrlist.add (45);  
        System.out.println ("ArrayList : " + Arrlist);  
        Object [] arr = Arrlist.toArray ();  
        System.out.println ("Elements of ArrayList"  
        " as array : " + Arrays.toString (arr));  
    }  
}
```

Q7) How do you retrieve an element from a particular position of an arraylist?

Get() method returns an element from a specified position of an arraylist. This method takes index of the element as an argument.

```

import java.util.ArrayList;
public class ArrayListA8
{
    ArrayList<Integer> arr = new ArrayList<int>();
    arr.add(10);
    arr.add(20);
    arr.add(30);
    arr.add(40);
    System.out.println("List:" + arr);
    // element at index 2
    int element = arr.get(2);
    System.out.println("the element at index
    is:" + element);
}

```

Q.10] How do you replaces a particular element in an ArrayList with the given element.

set() method replaces a particular element in our arraylist with the given element. This method takes two arguments, one is the index of the element to be replaced and another one is the element to be replaced at that position.

```

import java.util.ArrayList;
public class ArrayListEx {
    public static void main (String args[])
    {
        ArrayList < Integer> arr = new ArrayList < Integer> ();
        arr.add (10);
        arr.add (20);
        arr.add (30);
        arr.add (40);
        System.out.println ("List before adding
                           element :" + arr);
        // element at index 2
        int element = arr.set (3, 25);
        System.out.println ("List after adding elem
                           ent :" + arr);
    }
}

```

Q1] How do you append an element at the end of an arraylist ?

Add() method appends an element at the end of an arraylist .

```

import java.util.ArrayList;
public class ArrayListEx {
    public static void main (String args[])
    {
    }
}

```

```
{
    ArrayList<integer> arr = new ArrayList<integer>();
    arr.add(10);
    arr.add(20);
    arr.add(30);
    arr.add(40);
    System.out.println("List before adding element
        + arr");
    //element at index 2
    arr.add(50);
    System.out.println("list after adding element
        at the end "+arr);
}
}
```

Q12] How do you insert an element at a particular position of an arraylist?
 Add() method which takes index and an element as arguments can be used to insert an element at a particular position of an arraylist. The elements at the right side position are shifted one position right indices of right side elements of that position are increased by 1.

```
public class arrayListA11 {  
    public static void main (String args [])  
    {  
        ArrayList <integer> arr = new ArrayList  
        <integer> ();  
        arr.add (10);  
        arr.add (20);  
        arr.add (30);  
        arr.add (40);  
        System.out.println ("list before adding  
        element :" + arr);  
        arr.add (2, 35);  
        System.out.println ("list after adding element  
        at third Position " + arr);  
    }  
}
```

Q.13] How do you remove an element from a particular position of an arraylist?
Remove () method which takes int types as an argument is used to remove an element from a particular position of an arraylist.

```
import java.util.ArrayList;  
Public class arrayListA12 {
```

```
public static void main (String args [])  
{  
    ArrayList <Integer> arr = new ArrayList <Integer>()  
    ;  
    arr.add (100);  
    arr.add (200);  
    arr.add (300);  
    arr.add (400);  
    System.out.println ("List before removing  
    element : " + arr);  
    arr.remove (2);  
    System.out.println ("list after removing element  
    at third position " + arr);  
}
```

Q.14] How do you remove the given element from
an arraylist?

Remove (Object obj) method removes the first
occurrence of the specified element 'obj'. If
that element doesn't exist, arraylist will be
unchanged.

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

```
public class arrayListA {
```

```
{  
    public static void main (String args [])  
    {  
        ArrayList<String> arr = new ArrayList<>();  
        arr.add ("c");  
        arr.add ("c#");  
        arr.add ("c");  
        arr.add ("java");  
        System.out.println ("list before removing  
        element : "+arr);  
        arr.remove ("c");  
        System.out.println ("list after removing  
        element "+arr);  
    }  
}
```

Q.15] How do you remove all the elements of
an arraylist at a time?

clear() method removes all elements of
an arraylist : arraylist will be empty
after this method is executed.

Q.16] How do you retrieve a portion of an arraylist?

Using sublist() method of arraylist, we can retrieve a portion of an arraylist. sublist() method returns a view of a portion of an arraylist in the given range.

The returned sublist is backed by original arraylist. That

Means any changes made to sublist will be reflected in original arraylist or vice-versa.

```
import java.util.*;  
class arraylist A15  
{  
    public static void main (String args [])  
    {  
        ArrayList <Integer> arr = new ArrayList<Integer>();  
        arr.add(10);  
        arr.add(20);  
        arr.add(30);  
        arr.add(40);  
        List <Integer> arr2 = arr.sublist(1,3);  
        System.out.println ("sublist of arraylist:" + arr2)  
        System.out.println ("elements of arraylist:" +  
                           arr);  
    }  
}
```

Q.17] How do you join two ArrayList? We can use addAll() method which takes collection type as an argument to join two ArrayLists. This method appends all elements of the passed collection to the end of the invoking collection.

```
class ArrayListA16
{
    public static void main (String args[])
    {
        ArrayList < Integer> arr = new ArrayList <>()
        arr.add (2);
        arr.add (4);
        arr.add (6);
        arr.add (8);
        arr.add (10);
        System.out.println ("Elements of first
                            arraylist:" + arr);
        ArrayList < Integer> arr2 = new ArrayList <>()
        arr2.add (12);
        arr2.add (14);
        arr2.add (16);
        arr2.add (18);
        arr2.add (20);
        System.out.println ("Elements of sec
```

```
arr.addAll(arr2);
System.out.println("list1+list2 = " + arr);
}
}
```

Q.18] How do you insert more than one element at a particular position of an arraylist?
Another version of addAll() method which takes two arguments, one is index and another one is collection type can be used for this requirement. This method inserts all of the elements of passed collection at a specified position of an arraylist.

```
import java.util.*;
class arrayListA17
{
    public static void main (String args[])
    {
        ArrayList < Integer > arr = new ArrayList <> ();
        arr.add (15);
        arr.add (25);
        arr.add (85);
        arr.add (95);
        arr.add (105);
        System.out.println ("Elements of first array is
                            + arr);
```

```
ArrayList<Integer> arr2 = new ArrayList<>();
arr2.add(35);
arr2.add(45);
arr2.add(55);
arr2.add(65);
arr2.add(75);
System.out.println("Elements of second
arraylist : " + arr2);
arr.addAll(2, arr2);
System.out.println("list1+List2 = " + arr);
}
}
```

Q.19] write a program using all the methods described above.

```
import java.util.*;
class ArrayListA18
{
    public static void main (String args[])
    {
        ArrayList<String> arr = new ArrayList<>();
        arr.add ("Collection framework in Java");
        arr.add (1, "ArrayList");
        System.out.println ("Elements of first
list : " + arr);
```

```
System.out.println();
ArrayList<String> arr2 = new ArrayList<>();
arr2.add ("vector");
arr2.add ("list");
arr2.add ("Linked List");
arr2.add ("vector");
System.out.println ("Elements of second list:")
+ arr2);
System.out.println();
arr.addAll (2, arr2);
System.out.println ("list1+list2" + arr);
System.out.println ();
ArrayList<String> arr3 = new ArrayList<>();
arr3.add ("stack");
arr3.add ("Hashet");
System.out.println ("Elements of third list:")
+ arr3);
System.out.println ();
arr.addAll (arr3);
System.out.println ("list1+list3=" + arr);
System.out.println ();
boolean flag = arr.contains ("List");
if (flag == true) {
System.out.println ("alist contains
element list");
```

```
System.out.println();
} else {
    System.out.println(" a list doesn't contain
        element list");
    System.out.println();
}
String value = arr.get(2);
System.out.println("Element Retrieved at
    index 2 i.e. 3rd position = " + value);
System.out.println();
int value2 = arr.indexOf("Linked list");
System.out.println(" index Retrieved of
    linked list = " + value2);
System.out.println();
arr.ensureCapacity(20);
System.out.println("ArrayList numbers");
System.out.println();
int index1 = arr.lastIndexOf("vector");
System.out.println(" index of last vector
    in ArrayList: " + index1);
System.out.println();
arr.set(2, "Stack");
System.out.println("The arrayList after
    setting 333 value at index 2, " + arr);
System.out.println();
```

```
arr.remove(2);
System.out.println("ArrayList After removing
element at index 2 = "+arr);
System.out.println();
System.out.println("size of arraylist is: "+
arr.size());
System.out.println();
Object[] object = arr.toArray();
for(int i=0; i<object.length; i++){
    System.out.println("value at index " + i +
    " of array converted from ArrayList = " +
    object[i]);
}
System.out.println();
arr.trimToSize();
System.out.println("Printing an ArrayList
after using trimToSize method " + arr);
System.out.println();
boolean flag1 = arr.isEmpty();
if(flag1 == true){
    System.out.println("ArrayList is Empty");
}
else{
    System.out.println("ArrayList is not empty");
}
```

```
System.out.println();  
arr.clear();  
System.out.println("empty arraylist after  
using clear method;" + arr);  
}  
}
```