Assignment - 4

Page No.
Date

Q1. 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);
// creating arraylist using arraylist
   class
List<String> array2 = new ArrayList<>(4);
// creating 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 seco-
nd arraylist :" + array2);
  }
}
```

Elements of first arraylist : [1, 2, 3]
Elements of second arraylist : [Java, programming]

92. 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 arraylistex1 {
public static void main(String args[])
    {
// create an empty array list with
an initial capacity
ArrayList<Integer>(5);
// use add method to add elements
arrlist.add(10);
arrlist.add(50);
arrlist.add(30);
```

```
//this will increase the capacity of
the ArrayList to 15 elements.
arrlist.ensureCapacity(15);
// let us print all the elements ava-
ilable in list

for (Integer number : arraylist) {
System.out.println("Number =" + number)
    }
  }
}
```

output:
Number = 10
Number = 50
Number = 30

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 arraylistex2 {
public static void main (String args[])
{
ArrayList <Integer> arr2 = new Array-
List <>(5);
arr2.add (10);
arr2.add (20);
arr2.add (30);
arr2.add (40);
arr2.TrimToSize (); // trim the array-
list
System.out.println ("Elements in given
arraylist:" +arr2);
}
}
```

Output:

Elements in given arraylist: [10,20,30,40]

4. 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 arraylistex4
{
```

```
public static void
    {
ArrayList<String> arr2 = new Arra-
    ylist<>();
arr2.add("Java");
arr2.add("virtual");
arr2.add("machine");
System.out.println("Numbers of elem-
ents present in arraylist is =" +
    arr2.size());

System.out.println("Elements in
given arraylist=" + arr2);
    }
}
```

Output:

Number of elements present in array-list is = 3

Elements in given arraylist=[Java, virtual, machine]

95. 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 arraylistex5
{
public static void main(String args[])
{
Arraylist<Integer> arr2 = new Arraylist<>(5);
arr2.add(10);
arr2.add(20);
arr2.add(30);
arr2.add(40);
boolean result = arr2.isEmpty();
if (result == true)
{
System.out.println("arraylist is empty");
}
else
{
System.out.println("arraylist is not empty");
System.out.println("elements in the arraylist are:" + arr2); }
}
}
```

Output:
arraylist is not empty
elements in the arraylist are:
[10, 20, 30, 40]

Q6. 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 arraylistex6
{
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 search");
Scanner s = new Scanner(System.in);
int a = s.nextInt();
if (arr2.contains(a))
    {
System.out.println ("arraylist conte-
    ins " + a);
    }
  else
    {
System.out.println ("arraylist does
not contain " + a); }
        }
    }
```

output:

```
java arraylistex6
enter the element to search
    30
arraylist contains 30
```

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 first occurrence of a specified element whereas lastIndexOf() method returns index of last occurrence of a specified element in an arraylist. If element is found they will return -1.

```
import java.util.*;
class arraylistex7
{
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 find 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); }
}
```

output:

enter the element to Find index of it

30

index of 30 is 2

6. How do you convert an ArrayList to Array?

Using toArray() method of ArrayList class. to Array() 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 arraylistex8 {
public static void main (String args[])
    {
ArrayList<Integer> Arrlist = new
 ArrayList<Integer>();
  Arrlis.add(15);
  Arrlis.add(25);
  ArrLis.add(35);
  ArrLis.add(45);
System.out.println("Array List:" + Arr-
                    Lis);
```

```
Object[] arr = ArrLis.toArray();
System.out.println("Elements of
ArrayList " + " as Array: " + Arrays.
toString(arr));
        }
    }
```

output:

ArrayList : [15, 25, 35, 45]
Elements of Arraylist as Array:
[15, 25, 35, 45]

99. 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 arrayliste.g {
public static void main(String[]
                    args)
    {
ArrayList<Integer> arr = new Array-
List<Integer>();
    arr.add(10);
```

```
        arr.add(20);
        arr.add(30);
        arr.add(40);
System.out.println("List:"+arr);
//element at index 2 is
int element = arr.get(2);
System.out.println("the element
at index 2 is"+element);
    }
}
```

output:

List:[10, 20, 30, 40]

the element at index 2 is 30

Q10. How do you replaces a particular element in an ArrayList with the given element.

set() method replaces a particular element in an 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 placed at that position.

```
import java.util.ArrayList;
public class arraylistex10
{
```

```
public static void main (String args[])
    {
ArrayList<Integer> arr = new Array
List<Integer>();
    arr.add(10);
    arr.add(20);
    arr.add(30);
    arr.add(40);
System.out.println("List before add-
ing element:" + arr);
// element at index 2
  int element = arr.set(3,25);
System.out.println("list after add-
  ing element" + arr); }
    }
```

Output:

List before adding element: [10, 20, 30, 40]

list after adding element [10, 20, 30, 25]

Q.11. 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 arraylistex11 {
ArrayList<Integer> arr = new ArrayList<>();
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);
}
}
```

output:

List before adding element: [10, 20, 30, 40]
List after adding element at the end
[10, 20, 30, 40, 50]

Q12 How do you insert an element at a particular position ~~at a particular~~ of an ArrayList?

Add() method which takes index and and 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 i.e. indices of right side elements of that position are increased by 1.

```
import java.util.ArrayList;
public class arraylistex12
{
public static void main (String[]
        args) {
Arraylist <Integer> arr = new Array
List<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 add-
ing element at third position"+arr);
        }
    }
```

Q13. 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 static class arraylistex13 {
public static void main (String[] ar-
    gs) {
ArrayList<Integer> arr = new Array-
List<Integer>();
    arr.add(100);
    arr.add(200);
    arr.add(300);
    arr.add(400);
System.out.println("List before remo-
ving element:" + arr);

    arr.remove(2);
System.out.println("List after remo-
ving element at third position" +
    arr); } }
```

output:

List before removing element: [100, 200, 300, 400]

list after removing element at third position [100, 200, 400]

Q14. 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 arraylistex14 {
public static void main(String[] ar
        gs)
    {
ArrayList<String> arr = new
    ArrayList<>();
    arr.add("c");
    arr.add("c++");
    arr.add("java");
    arr.add("ada");
System.out.println("List before remo
ving element:" + arr);
    arr.remove("c");
System.out.println("List after re
moving element " + arr); }
}
```

output :

List before removing element: [c, c++, java, oda]

list after removing element [c++, java, ada]

Q15. How do you remove all the elements of an ArrayList at a time?

clear() method removes all elements of an Arraylist. Arratlist will be empty after this method is excuted.

Page No.
Date

Q16. 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 arraylister16
    {
public static void main(String args[])
    {
ArrayList<Integer> arr=new Array-
List<>();
    arr.add(56);
    arr.add(67);
    arr.add(78);
    arr.add(89);
List(Integer> arr2=arr.sublist(1,3)
System.out.println("sublist of array-
```

System.out.println("elements of arraylist :" +arr);
}

output:

sublist of arraylist: [67, 78]
elements of arraylist: [56, 67, 78, 89]

Q.17 How do you join two Arraylists?

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.

```
import java.util.*;
class arraylistex17
{
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 Array-
    List<>();
    arr2.add(12);
    arr2.add(14);
    arr2.add(16);
    arr2.add(18);
    arr2.add(20);
System.out.println("Elements of second
        arraylist:" + arr2);
    arr.addAll(arr2);
System.out.println("List1 + List2 ="
        + arr);
    }
}
```

Output:

Elements of first arraylist: [2, 4, 6, 8, 10]
Elements of second arraylist: [12, 14, 16, 18, 20]
List1 + List2 = [2, 4, 6, 8, 10, 12, 14, 16, 18, 20]

Q18. 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 arraylistex18 {
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 arraylist:" + 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);
    }
}
```

output:

Elements of first arraylist: [15, 25, 85, 95, 105]

Elements of second arraylist: [35, 45, 55, 65, 75]

List1 + list2 = [15, 25, 35, 45, 55, 65, 75, 85, 95, 105]

9.19. write a program using all the methods described above.

```
import java.util.*;
class arraylistex19 {
public static void main(String args[])
{
```

```
ArrayList<String> arr = new Arra
List<>();
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("list1 + list2"
      + arr);
System.out.println();
ArrayList<String>
  arr3 = new ArrayList<>();
  arr3.add("stack");
  arr3.add("Hashset");
```

```
System.out.println("Elements of
third list :" +arr3);
System.out.println();
arr.addAll(arr3);
System.out.println("List 1 + list 3="
+arr);
System.out.println();
boolean Flag = arr.contains("list");
if (Flag == true) {
system.out.println("a list contains
element list");

system.out.println();
}
else
{
System.out.println("a list doesn't
contains element list");
system.out.println();
}

String value = arr.get(2);
System.out.println("Element Retri-
eved 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("Array List Number =" +arr);
System.out.println();
    int index1 = arr.lastIndexOf("vector");
System.out.println("index of last vector in Arraylist:" +index1);
System.out.println("The Arraylist after setting 333 value at index2
        +arr);
system.out.println();

  arr.remove(2);
System.out.println("Array List After Removing Element at index 2="
    +arr);
System.out.println();
```

```
System.out.println("Size of Array-
list 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 Coverted 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("Array List is
    Empty");
      }
      else{
System.out.println("Array List is not
```

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

output:
Elements of first list : [collection framework in java : , Arraylist]
Elements of second list : [vector , list , linked list , vector]
List 1 + List 2 - [collection framework in java : , Arraylist , vector , list , linked list , vector]

Elements of third list : [Stack , Hashset]
list 1 + List 3 = [collection framework in java : , Arraylist , vector , list , linked list , vector , stack , Hashset]

a list contains element list
Element Retrived at index 2 i.e. 3rd position = vector
Index Retrived of linked list = 4
Array List Number 2 [collection framework in java : , Arraylist , vector , list , linked list , vector , stack ,

index of last vector in Arraylist :5
The Arraylist after setting 333 value at index 2 : [collection framework in java :, Arraylist, stack, list, linked list, vector, stack, Hashset]
Arraylist After Removing Element at index 2 = [collection framework in java:, Arraylist, list, linked list, vector, stack, Hashset]

size of Arraylist is: 7
value at index 0 of Array converted From Arraylist = collection framework in java:
value at index 1 of Array converted from Arraylist = Arraylist

empty arraylist after using clear method : []
Elements of first list : [collection framework in java :, Arraylist]
Elements of second list : [vector, list, linked list, vector]

List 1 + List 2 = [collection framework in java :, Arraylist, vector, list, linked list, vector]

Elements of third list : [stack, Hash
list 1 + list 3 = [collection framework
in java :, Arraylist, vector, List,
Linked List, vector, stack, Hashset

a list contains element list
Element retrived at index 2 i.e.
3rd position = vector
Index Retrived of linked list = 4
Array list Number = [collection frame-
work in java :, Arraylist, vector, list,
Linked list, vector stack, Hash set]

Index of Last vector in Arraylist is
The Arraylist after setting 333
value at index 2 : [collection frame-
work in java :, Arraylist, stack,
list, Linked list, vector, stack,
Hashset]
Arraylist After Removing Element
at index 2 = [collection framework in
java :, Arraylist, List, Linked List,
vector, stack, Hashset]

size of Arraylist is : 7
value at index 0 of Array converted

Page No.
Date

From Arraylist = collection frame-work in java:
value at index 1 of array converted From Arraylist = Arraylist.