VelammalCollege of Engineering and Technology, Madurai
Department of Computer Science and Engineering
Object oriented programming lab

Exercise 7
ArrayList

Team 1

1. Write a Java program to perform the following task.
   Take an integer array of size 20, initialize values randomly between 10 and 90, simultaneously sum all values and calculate average. Now separate values below average and above average in ArrayLists. Finally print both lists in 2 separate rows.

2. Create a student class with name,rollno,cgpa as attributes. Store 20 students information in arraylist. Use Comparator interface to sort the students details according to cgpa.

Team 2

1. Write a program to perform string operations using ArrayList. Write functions for the following
   a. Append - add at end
   b. Insert – add at particular index
   c. Search
   d. List all string starts with given letter
2. Create a student class with name,rollno,cgpa as attributes. Store 20 students information in arraylist. Use Comparator interface to sort the students details according to rollno.

Team 3

1. Write a program in java that takes as input the telephone bill amount for 12 months and calculate average telephone bill in the year. The program should display least and maximum bill amounts.

E,g)

```
Enter phone nos for 12 months
120.0
230.0
100
100
100
100
50
50
50
50
30
30

Average:84.1666
Least bill:30
```

Maximum bill:230

2. Create a student class with name,rollno,cgpa as attributes. Store 20 students information in arraylist. Use Comparator interface to sort the names in ascending order.

Team 4

1. Write a program that reads a set of string and displays the words in a list.

First display all words.

display them with all plurals (ending in "s") capitalized.

display them in reverse order.

display them with all plural words removed.

2. Create a student class with name,rollno,cgpa as attributes. Store 20 students information in arraylist. Use Comparator interface to sort the names in descending order.

Team 5

1. Write a program that reads numbers and displays all the numbers as a list, then:

Prints the average of the numbers.

Prints the highest and lowest number.

Filters out all of the even numbers (ones divisible by 2).

2. Create a employee class with name,salary as attributes. Store 20 employees information in arraylist. Use Comparator interface to sort the names in descending order.

Team 6

1. Write a method intersect that accepts two sorted array lists of integers as parameters and returns a new list that contains only the elements that are found in both lists.

   Example: if lists named list1 and list2 initially store:

   [1, 4, 8, 9, 11, 15, 17, 28, 41, 59]

   [4, 7, 11, 17, 19, 20, 23, 28, 37, 59, 81]

   Then the call of intersect(list1, list2) returns the list: [4, 11, 17, 28, 59]

2. Create a employee class with name,salary as attributes. Store 20 employees information in arraylist. Use Comparator interface to sort the names in ascending order.

Team 7

1. Design a Java console application for implement string operations - Search a string in a list, and Display all the strings that begin with a given letter and end with a letter using ArrayList in java.

2. Create a employee class with name,salary as attributes. Store 20 employees information in arraylist. Use Comparator interface to sort the salary in ascending order.

Comparator Interface

Comparator interface is used to order the objects of user-defined classes. A comparator object is capable of comparing two objects of two different classes. Following function compare obj1 with obj2 public int compare(Object obj1, Object obj2):

How does Collections.Sort() work?

Internally the Sort method does call Compare method of the classes it is sorting. To compare two elements, it asks "Which is greater?" Compare method returns -1, 0 or 1 to say if it is less than, equal, or greater to the other. It uses this result to then determine if they should be swapped for its sort.

public void sort(List list, ComparatorClass c)

Steps:

1. Create ArrayList for storing objects of a class
2. Create class to compare two object, which implements Comparator Interface

```
class Sortbyname implements Comparator<Student>
 {
 public int compare(Student a, Student b)
 {
    return a.name.compareTo(b.name);
 }
}
```

3. Use Collections class to sort the object according to some constraint.
   Collections.sort(ar, new Sortbyroll());