

1. Create a class Employee with three data members as given below:

```
class Employee{  
    int empId;  
    String empName;  
    double salary;  
}
```

Create an ArrayList of the above-mentioned class and perform the below operations on the same.

- a. Add Employee objects in the list.
  - b. Remove the object at a particular Index.
  - c. Print the list using the below methods:
    - i. Using for loop
    - ii. Using for-each loop
    - iii. Using Iterator Interface
    - iv. Print backward using ListIterator interface.
  - d. Remove all the instances from the list who have salary less than 10000. Use Iterator interface to perform the functionality.
2. Using the above class, what will be the output of the below mentioned code snippet? Share your findings with the help of comments. What should be done to resolve the problem if any?

```
List<Employee> empList = new ArrayList<Employee>();  
  
empList.add(new Employee(5,"A",45000));  
empList.add(new Employee(2,"B", 30000));  
empList.add(new Employee(7,"C", 80000));  
  
boolean check = empList.contains(new Employee(5,"A",45000));  
System.out.print(check);  
  
empList.remove(new Employee(5,"A",45000));  
  
System.out.print(empList.size());
```

3. Sort the above list on empId using Comparable interface.
4. Sort the same list again on basis of salary using Comparator Interface. Show the implementation of Comparator with the help of
  - a. Anonymous Inner Class
  - b. Lambda Expression
5. Change the above code to use LinkedList instead of ArrayList and perform the same operations.

6. Create a Java program that creates a Set using a HashSet implementation. The Set stores fruit names and the fruits must be added in the following sequence:

Pear  
Banana  
Tangerine  
Strawberry  
Blackberry

The functionalities which must be exhibited by the program are:

- a. Display the contents of set.
- b. Display the number of elements in the set.
- c. Remove blackberry and strawberry, display contents of set again.
- d. Remove the remaining fruits using a single method invocation.
- e. Show the set is now empty.

→ While displaying the set contents, explain why the fruits don't get listed in the order in which they were added.

7. Create a HashSet of the class Employee created in Question-1.

```
HashSet<Employee> employeeSet = new HashSet<Employee>();  
employeeSet.add(new Employee(1, "A", 45000));  
employeeSet.add(new Employee(5, "D", 55000));  
employeeSet.add(new Employee(7, "B", 35000));  
employeeSet.add(new Employee(2, "P", 75000));  
employeeSet.add(new Employee(1, "A", 45000));  
  
System.out.println(employeeSet.size());
```

Check the result and document your findings in a form of comments.

Print the HashSet with the help of Iterator Interface. Explain why HashSet can't be index-based. Create a readme.txt file to document your answer.