Java Programming Assignment-12

Part A-Theory Questions

1)What is Linked List?

Ans: In Java Collection Framework, LinkedList is a class used to store data in linked-list form.

LinkedList stores elements as nodes.

Each node has data + reference to next (and previous) node.

It is dynamic (size can grow or shrink).

Where it belongs

Collection

↓

List

↓

LinkedList

LinkedList implements List interface.

Also implements Deque (double-ended queue).

Elements are not stored in continuous memory.

Allows duplicate values.

Maintains insertion order.

Slower access (get()), faster insert/delete.

2)Difference between ArrayList and LinkedList.

Ans:

Feature	**ArrayList**	**LinkedList**
Internal structure	Uses dynamic array.	Uses doubly linked list.
Data storage (data +	Elements stored in continuous memory.	Elements stored as nodes +address).

Access (get)	Fast (direct index access)	Slow (traverses node by node).
Insertion / Deletion	Slow (shifting required).	Fast (no shifting).
Memory usage	Uses less memory	Uses more memory
Search performance	Faster.	Slower.
Best for	Reading/searching data.	Frequent insert/delete operations.

3)What is HashSet?

Ans:

HashSet is a collection in Java that:

Stores unique elements only.

Does not allow duplicates.

Does not maintain insertion order.

It is part of the Set interface.

Key points of HashSet:

1)No duplicate values.

2) No order (random order).

3)Allows one null value.

4) Fast searching (uses hashing).

5)No index-based access (no get().

HashSet stores unique elements in random order using hashing.

Example:

```
import java.util.HashSet;

class Test {

   public static void main(String[] args) {

      HashSet<Integer> hs = new HashSet<>();

      hs.add(10);
```

```
        hs.add(20);

        hs.add(10);

        hs.add(30);

        System.out.println(hs);

    }

}
```

O/P:[20, 10, 30]

10 is printed only once because duplicates are not allowed.

4)What is LinkedHashSet?

Ans:

LinkedHashSet stores unique elements and keeps the order in which elements are added.

LinkedHashSet is a Set in Java that:

Stores unique elements.

Maintains insertion order.

Does not allow duplicates.

It is a child class of HashSet.

Key points of LinkedHashSet:

1)No duplicate elements.

2)Maintains insertion order.

3) Allows only one null value.

4) Faster than TreeSet.

5) No index-based access.

Example:

```
import java.util.LinkedHashSet;

class Test {

    public static void main(String[] args) {
        LinkedHashSet<Integer> lhs = new LinkedHashSet<>();
        lhs.add(10);
        lhs.add(20);
        lhs.add(10);
        lhs.add(30);
        System.out.println(lhs);
    }
}
```

o/p:[10, 20, 30]

Order is same as insertion order.

5)What is TreeSet?

Ans:

TreeSet is a Java Set that stores only unique elements and automatically arranges them in sorted (ascending) order.

TreeSet is a Set in Java that:

Stores unique elements.

Keeps elements in sorted order.

Does not allow duplicates.

It implements the SortedSet / NavigableSet interface.

Key Points of TreeSet:

1)No duplicate elements.

2) Sorted order (ascending by default).

3) No insertion order.

4)Does not allow null (throws NullPointerException).

5) Slower than HashSet & LinkedHashSet.

Example:

```
import java.util.TreeSet;

class Test {

    public static void main(String[] args) {

        TreeSet<Integer> ts = new TreeSet<>();

        ts.add(30);

        ts.add(10);

        ts.add(20);

        ts.add(10);

        System.out.println(ts);

    }

}
```

o/p:[10, 20, 30]

Automatically sorted.

Duplicate 10 is ignored.

Part B- Practical Questions

1)Write a program to reverse a LinkedList.

Ans:

```
import java.util.Iterator;

import java.util.LinkedList;

import java.util.Scanner;


public class ReverseLinkedList {

        public static void main(String[] args) {

                LinkedList<Integer> list= new LinkedList<Integer>();
```

```
		Scanner sc = new Scanner(System.in);
		System.out.println("Enter how many numbers:");
		int n=sc.nextInt();
		System.out.println("enter a element");
		for (int i = 0; i < n; i++) {
			list.add(sc.nextInt());
		}
		System.out.println("Reverse Number");
		Iterator<Integer> itr=list.descendingIterator();
		while(itr.hasNext()) {
			System.out.println(itr.next());
		}
	}
}
```

2)Write a Program to iterate ArrayList using Iterator.

Ans:

```
import java.util.ArrayList;
import java.util.Iterator;
import java.util.Scanner;
public class ArrayList1 {
	public static void main(String[] args) {
		// TODO Auto-generated method stub
		ArrayList<Integer> arr = new ArrayList<Integer>();
		Scanner sc = new Scanner(System.in);
		System.out.println("Enter size:");
		int n=sc.nextInt();
```

```
        System.out.println("Enter numbers:");
        for(int i=0;i<n;i++) {
            arr.add(sc.nextInt());
        }
        System.out.println("Numbers are:");
        Iterator<Integer> itr=arr.iterator();
        while(itr.hasNext()) {
            System.out.println(itr.next());
        }
    }
}
```

3)Write a programs to sort an ArrayList of intgers.

Ans:

```
import java.util.ArrayList;
import java.util.Collections;
import java.util.Scanner;
public class ArrayListSort {
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        ArrayList<Integer> arr = new ArrayList<Integer>();
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the size:");
        int n=sc.nextInt();
        System.out.println("Enter the element:");
        for (int i = 0; i < n; i++) {
            arr.add(sc.nextInt());
        }
        System.out.println("Before sorting:"+arr);
        Collections.sort(arr);
        System.out.println("After sorting:"+arr);
    }
}
```

4)Write a program to remove duplicates from ArrayList.

Ans:

```
import java.util.ArrayList;
import java.util.Scanner;
import java.util.TreeSet;
public class RemoveDuplicateFromArrayList {
    public static void main(String[] args) {
        // TODO Auto-generated method stub
                ArrayList<Integer> arr = new ArrayList<Integer>();

                Scanner sc = new Scanner(System.in);
                System.out.println("Enter the size:");
                int n=sc.nextInt();
                System.out.println("Enter the element:");
                for (int i = 0; i < n; i++) {
                    arr.add(sc.nextInt());
                }
                System.out.println("List after sorting:");
                TreeSet<Integer> tree = new TreeSet<Integer>(arr);
                System.out.println(tree);
```

```
            }
}
```

5)Write a program to merge two ArrayList.

Ans:

```
import java.util.ArrayList;
public class MergeArrayList {
    public static void main(String[] args) {
        ArrayList<Integer> list1 = new ArrayList<Integer>();
        list1.add(10);
        list1.add(20);
        list1.add(30);
        ArrayList<Integer> list2 = new ArrayList<Integer>();
        list2.add(40);
        list2.add(50);
        list1.addAll(list2);
        System.out.println("Merged ArrayList: " + list1);
    }
}
```