Collection Framework

Assignment Questions

Assignment Questions

1. **What is the Collection framework in Java?**

   Ans :

- **The collection framework is introduce to overcome the problems of the Array**
- **There is the many problem to store the data in Array are fixed in size and Array except only homogeneous type of data.**
- **The collection framework is the collection of various interfaces and classes which is connected to each other.**
- **The collection is used to store the data efficiently the collection are dynamic in size.**
- **There are various inbuilt function in the collection framework to add, delete, fetch the data.**

2. **What is the difference between ArrayList and LinkedList?**

   Ans :

ArrayList	LinkedList
1. The ArrayList implements the list interface.	1. The LinkedList implements List interface.
2. The ArrayList is not memory efficient.	2. The LinkedList is memory efficient.
3. Direct insertion of data at any index is not efficient in ArrayList.	3. Direct insertion of data at any index is very efficient in the LinkedList.
4. The data stored in the continuous manner in the memory.	4. The data stored in the Discrete manner in the memory.
5. Ex:-	5. Ex:-

```java
import java.util.*;
public class CollectionArrayList {
    public static void
main(String[]args){
        ArrayList ar = new
ArrayList();
        ar.add(23);
        ar.add(53);
        ar.add("varshab");
        ar.add(90);
        ar.add('c');
        System.out.println(ar);
    }
}
```

Output:-

[23, 53, varshab, 90, c]

```java
import java.util.*;
public class DemoLinkedList {
    public static void
main(String[]args){
        LinkedList l = new
LinkedList();
        l.add(34);
        l.add(90);
        l.add("Chandel");
        l.add(12);
        l.add('v');
        System.out.println(l);
    }
}
```

Output:-

[34, 90, Chandel, 12, v]

3. What is the difference between Iterator and ListIterator?

Ans :

Iterator

1. The Iterator is used to fetch the data from the collection.
2. Iterator is used for all the classes present inside the collection to fetch the data.
3. The Iterator can fetch the data from first index to last index.
4. Iterator has the hasNext() command to check the data is present or not at the next index.
5. There is the next() command to go the next index of the collection.
6. Ex:-

```
import java.util.*;
public class DemoIterator {
    public static void
main(String[]args){
        ArrayList ar = new
ArrayList ();
        ar.add(10);
        ar.add(20);
        ar.add(30);
        ar.add(40);
        System.out.println(ar);
        Iterator it =
ar.iterator();
        while (it.hasNext()){

System.out.println(it.next());
        }
    }
}
```

Output:-

ListIterator

1. The ListIterator is also used to fetch the data from the collection.
2. ListIterator are only used in the classes which are implementing The List interface.
3. ListIterator can fetch the data from Last index to the first index that is in the reverse direction.
4. ListIterator has the hasPrevious() command to check the data is present at previous index or not.
5. There is the previous() command to go to the previous index of the collection.
6. Ex:-

```
import java.util.*;
public class DemoListIterator {
    public static void main
(String[]args){
        ArrayList ar = new ArrayList
();
        ar.add(10);
        ar.add(20);
        ar.add(30);
        ar.add(40);
        System.out.println(ar);
        ListIterator lit =
ar.listIterator(ar.size());
        while (lit.hasPrevious()){

System.out.println(lit.previous());
        }

    }
}
```

Output:-

[10, 20, 30, 40] 10 20 30 40	[10, 20, 30, 40] 40 30 20 10

4. What is the difference between Iterator and Enumeration?

Ans :

Iterator	Enumeration
1. Iterator is used to fetch the data from the collection.	1. Enumeration is used to fetch the data from legacy classes.
2. Iterator is used in Java when a collection is introduced.	2. Enumeration is used in java when the collection is not introduced.
3. Iterator is more efficient.	3. Enumeration is less efficient.
4. Iterator is used in all the classes of collection.	4. Enumeration is used in only two classes of the collection that are legacy classes vector and stack.
5. Ex:-	5. Ex:-

```
import java.util.*;
public class DemoIterator {
    public static void
main(String[]args){
        ArrayList ar = new
ArrayList ();
        ar.add(10);
        ar.add(20);
        ar.add(30);
        ar.add(40);
        System.out.println(ar);
        Iterator it =
ar.iterator();
        while (it.hasNext()){

System.out.println(it.next());
```

```
import java.util.*;
public class DemoEnumeration {
    public static void
main(String[]args){
        Vector v = new Vector();
        v.add(10);
        v.add(20);
        v.add(30);
        v.add(40);
        v.add(50);
        v.add(60);

        System.out.println(v);
        Enumeration e =
v.elements();
        while
```

```
        }
    }
}
```

Output :-

[10, 20, 30, 40, 50, 60]
10
20
30
40
50
60

```
(e.hasMoreElements()){

System.out.println(e.nextElement
());
        }
    }
}
```

Output :-

[10, 20, 30, 40, 50, 60]
10
20
30
40
50
60

5. What is the difference between List and Set?

Ans :

List	Set
1. List is the interface which extends the collection interface.	1. Set is the interface which extends the collection interface.
2. List contains the three classes that are ArrayList, LinkedList and the vector.	2. The set contains the two classes HashSet and LinkedHashSet which implement the Set interface.
3. The List is used to store the data in their classes.	3. The Set is used to store the data in their classes.

6. What is the difference between HashSet and TreeSet?

Ans :

HashSet

1. The HashSet is the class which implements the set Interface.
2. The HashSet is used to store the data in the collection.
3. The HashSet follows the Hash function and Hash table.
4. The HashSet stores the data in the buckets of HashSet.
5. The order of insertion is not preserved in HashSet.
6. The Heterogeneous type of data is allowed in the HashSet.
7. The duplicates are not allowed in HashSet.
8. Ex:-

```
import java.util.*;
public class DemoHashSet {
    public static void main
(String[]args){
        HashSet h = new
HashSet();
        h.add(10);
        h.add(20);
        h.add(30);
        h.add(40);
        h.add(10);
        h.add("varshab");

        System.out.println(h);
    }
}
```

Output:-

[20, 40, 10, varshab, 30]

TreeSet

1. The TreeSet is the class which implements the SortedSet interface.
2. The TreeSet is used to store the data in the collection.
3. The TreeSet follows the binary search tree data structure.
4. The TreeSet stores the data in the nodes of the binary search tree.
5. The order of insertion is not preserve in the TreeSet.
6. The Heterogeneous type of data is not allowed in TreeSet.
7. The duplicates are not allowed in TreeSet.
8. Ex:-

```
import java.util.*;
public class DemoTreeSet{
    public static void
main(String[]args){
        TreeSet ts = new TreeSet();
        ts.add(10);
        ts.add(20);
        ts.add(30);
        ts.add(40);
        ts.add(50);
        System.out.println(ts);
    }
}
```

Output:-

[10, 20, 30, 40, 50]

7. What is the difference between Array and ArrayList?

Ans :

Array

1. The Array is used to store the large data.
2. The Array are Fixed in size once the size is declared.
3. The Array can accept only homogenous type of data.
4. The Array is faster.
5. We can not add any specific data at any specific location.
6. There are no inbuilt function to fetch the data from the Array.
7. Ex:-

```
public class DemoArray {
    public static void main
(String[]args){
        int ar [] = new
int[3];
        ar[0] = 1;
        ar[1] = 2;
        ar[2] = 3;

        for(int i=0;
i<ar.length; i++){

System.out.println(ar[i]);
        }
    }
}
```

Output:-

1
2
3

ArrayList

1. The ArrayList is used to store large data
2. The ArrayList is Dynamic in size it can increase whenever we want.
3. The ArrayList can accept Homogenous as well as heterogenous type of data.
4. The ArrayList is slower as compared.
5. We can add the specific data at any location.
6. There are the inbuilt function to fetch the data from the ArrayList.
7. Ex:-

```
import java.util.*;
public class DemoArrayList {
    public static void main
(String[]args){
        ArrayList ar = new
ArrayList();
        ar.add(10);
        ar.add(20);
        ar.add(30);
        ar.add(40);
        ar.add(50);
        System.out.println(ar);
    }
}
```

Output:-

[10, 20, 30, 40, 50]