**15. Collections Framework**

• **Theory:**

1. **Introduction to Collections Framework:-** The Collections Framework in Java is a unified architecture designed to store, manipulate, and retrieve groups of objects efficiently. It provides various data structures and algorithms that simplify programming by offering a standard way to manage collections of objects.

**2.List, Set, Map, and Queue Interfaces:-**

**List**:- Represents an ordered collection (also known as a sequence) that allows duplicate elements. The order of elements is determined by their insertion order.

**Map:-** Represents a collection that does not allow duplicate elements. The order of elements is not guaranteed (except in specific implementations).

**Map:-**Represents a collection of key-value pairs, where each key is unique. A map does not allow duplicate keys but allows duplicate values.

**Interfaces** **:-**Represents a collection designed for holding elements prior to processing. Typically follows the First-In-First-Out (FIFO) principle, though other ordering methods exist.

**3.ArrayList, LinkedList, HashSet, TreeSet, HashMap, TreeMap.**

* **ArrayList:-** A resizable array implementation of the List interface that allows dynamic resizing, fast random access, and maintaining the order of elements. It allows duplicate elements.
* **LinkedList:-** A doubly-linked list implementation of the List and Deque interfaces, allowing for efficient insertions and deletions from both ends. It maintains the order of elements and allows duplicate elements.
* **HashSet:-** A set implementation that uses a hash table for storage, which does not allow duplicate elements and does not guarantee any specific order of elements.
* **TreeSet:-** A set implementation that uses a red-black tree, storing elements in a sorted order. It does not allow duplicate elements.
* **HashMap:-** A hash table-based implementation of the Map interface that stores key-value pairs. It allows for fast retrieval of values based on their unique keys and does not maintain any order.
* **TreeMap:-** A Map implementation that uses a red-black tree to store key-value pairs in sorted order based on the keys. It does not allow duplicate keys.

**4.Iterators and ListIterators:-**

* **Iterator:-** Iterator is a standard interface for iterating over a collection. It provides methods to traverse the collection in a sequential manner.
* **ListIterator:-** ListIterator is a more specialized interface that extends the Iterator interface and is specifically designed for List collections. It allows bidirectional traversal and provides additional functionality.