Difference between HashSet and Hashtable

HashSet: Implements the Set interface, stores unique elements only, and does not allow duplicates. It is not synchronized and allows one null element. Used when you want to store unique values without caring about order.

Hashtable: Implements the Map interface, stores key-value pairs, does not allow null keys or null values, and is synchronized (thread-safe). Used when you want a thread-safe key-value store.

Vector vs Stack

Vector: A growable array of objects that is synchronized. It can store any type of object and allows random access by index.

Stack: A subclass of Vector that implements a Last-In-First-Out (LIFO) data structure. It has additional methods like push() and pop() for stack operations.

Distinguishing factor: All Stacks are Vectors, but not all Vectors are Stacks. Stack adds specific LIFO behavior to Vector.

TreeSet

TreeSet is a NavigableSet implementation based on a TreeMap. It stores elements in sorted (ascending) order and does not allow duplicates. It provides log(n) time complexity for basic operations like add, remove, and search. It can also provide custom ordering through a Comparator.

Benefits of Each Structure

HashSet: Fast lookups and insertions (O(1) average), no duplicates, good for quick membership checks.

Hashtable: Thread-safe key-value store, prevents nulls, suitable for concurrent environments.

Vector: Synchronized, dynamic array that allows random access.

Stack: Simple LIFO data management, easy to implement undo/redo operations.

TreeSet: Maintains sorted order, fast navigation methods, useful for range queries.