

**Disadvantages of ArrayList:**

**Data entry:** If new data is added or removed form ArrayList then its data needs to be shifted to update the list. If ArrayList has n numbers of data’s then an add or removes will take O(n) times.

**Memory**: Data’s in the ArrayList is stored sequentially in the memory so for larger list will need significant contiguous blocks of memory.

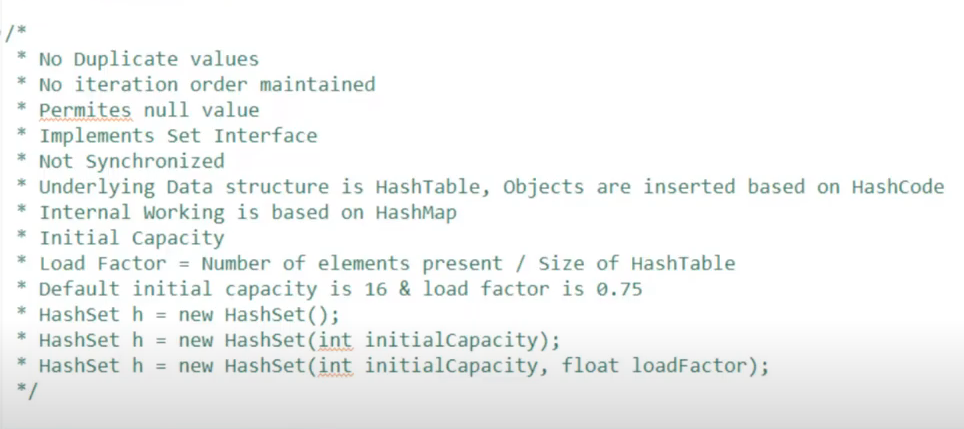
**Capcity**: ArrayList also has performance limitation, If you don’t specify initial capacity then by default it allocates it initial capacity which is 10. Every time when ArrayList hits its own capacity, data will be copy from old to new space with 50% more capacity.

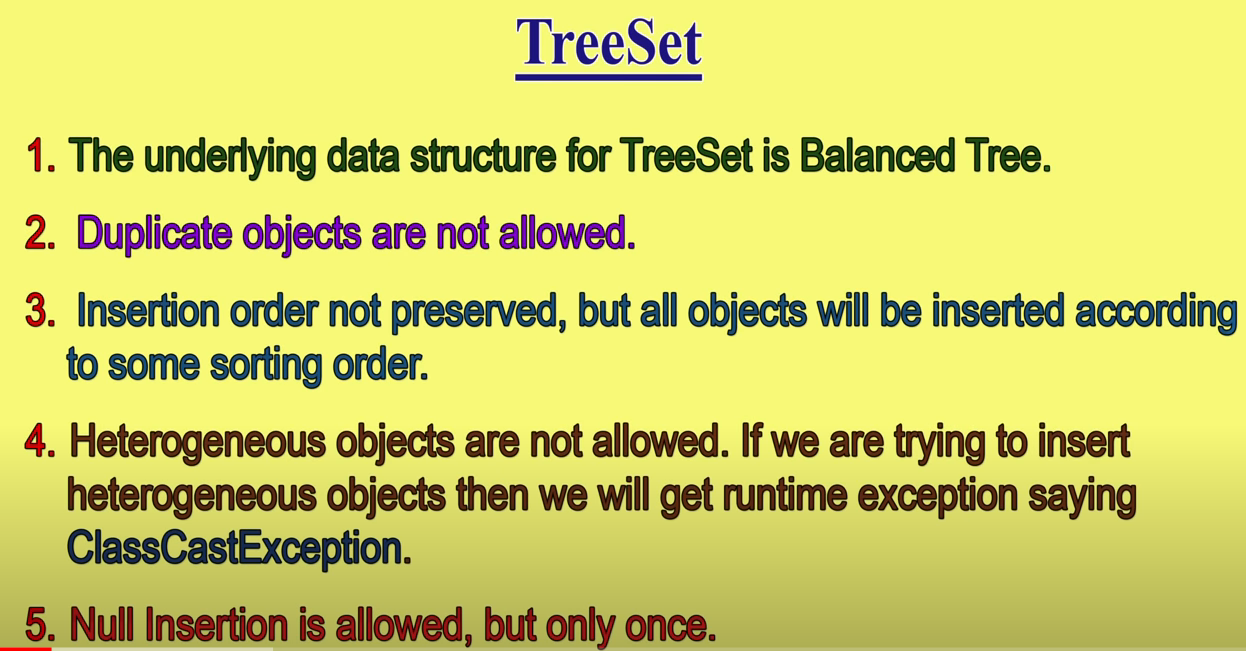
**Boxing and Unboxing**: ArrayList is not stronly typed as its loosly typed so boxing and unboxing takes place which hits its performance

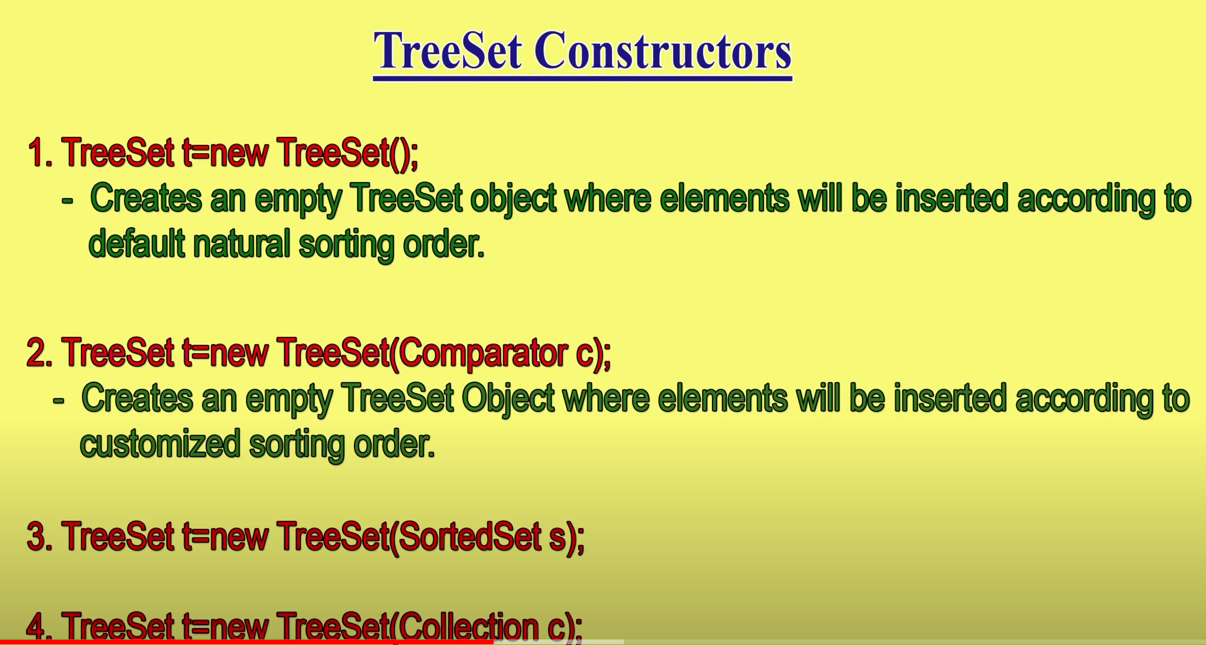
**Advantages of ArrayList:**

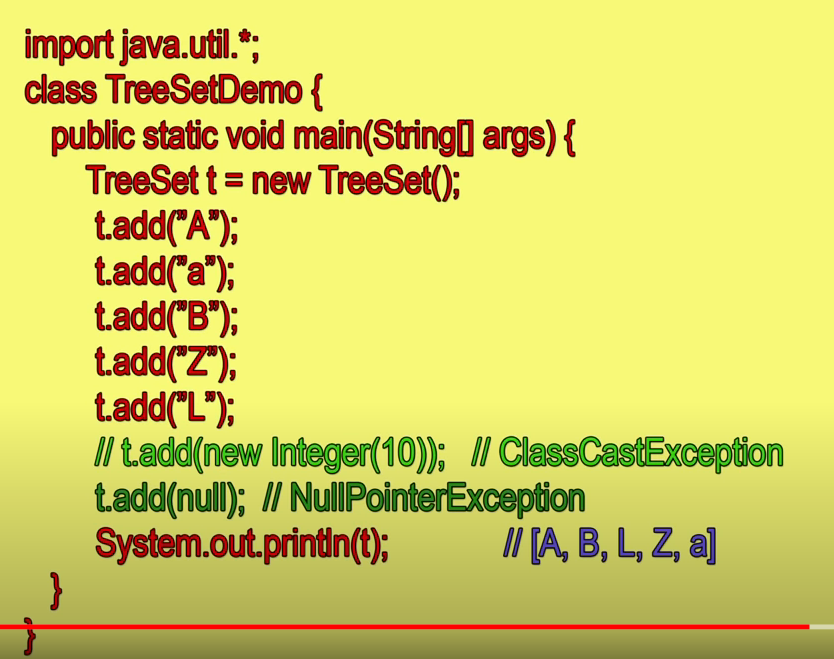
There are major advantages to ArrayLists when real-world projects are concerned:

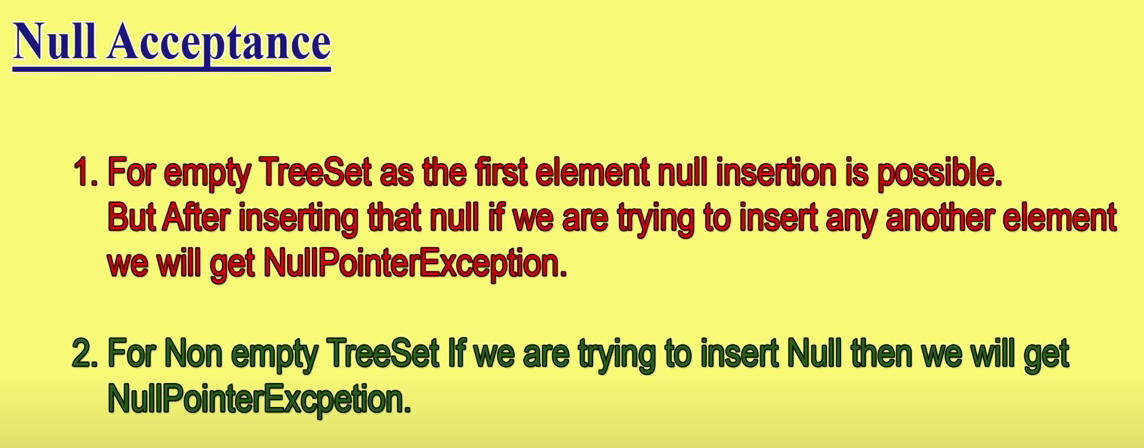
* ArrayLists can be appended dynamically: ArrayLists do not have to have a definite memory allocation like normal arrays when they are declared, they can be appended upon runtime. This saves unnecessary memory usage by the program.
* The insertion and searching implementations of ArrayLists are stronger than normal arrays and provide more efficiency.

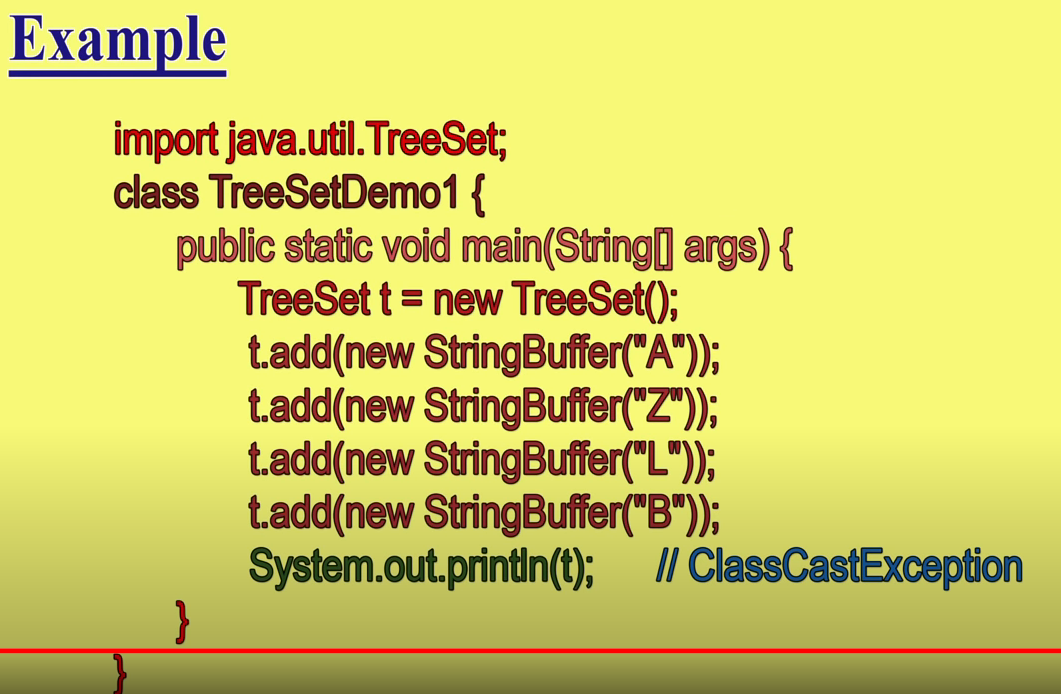


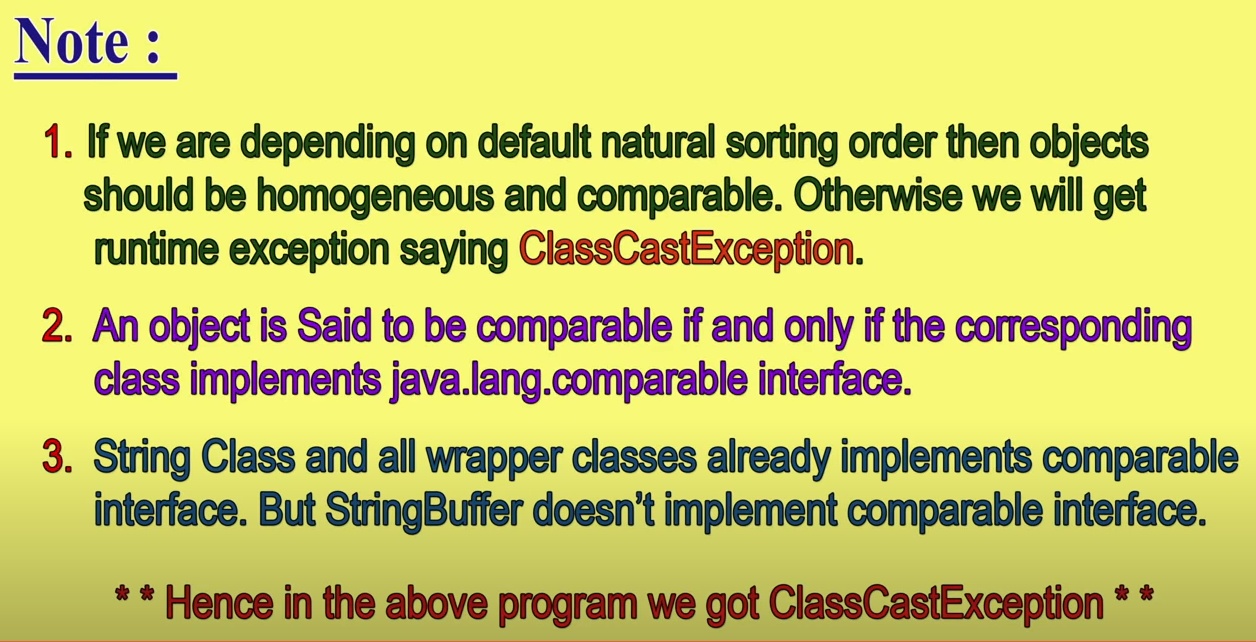


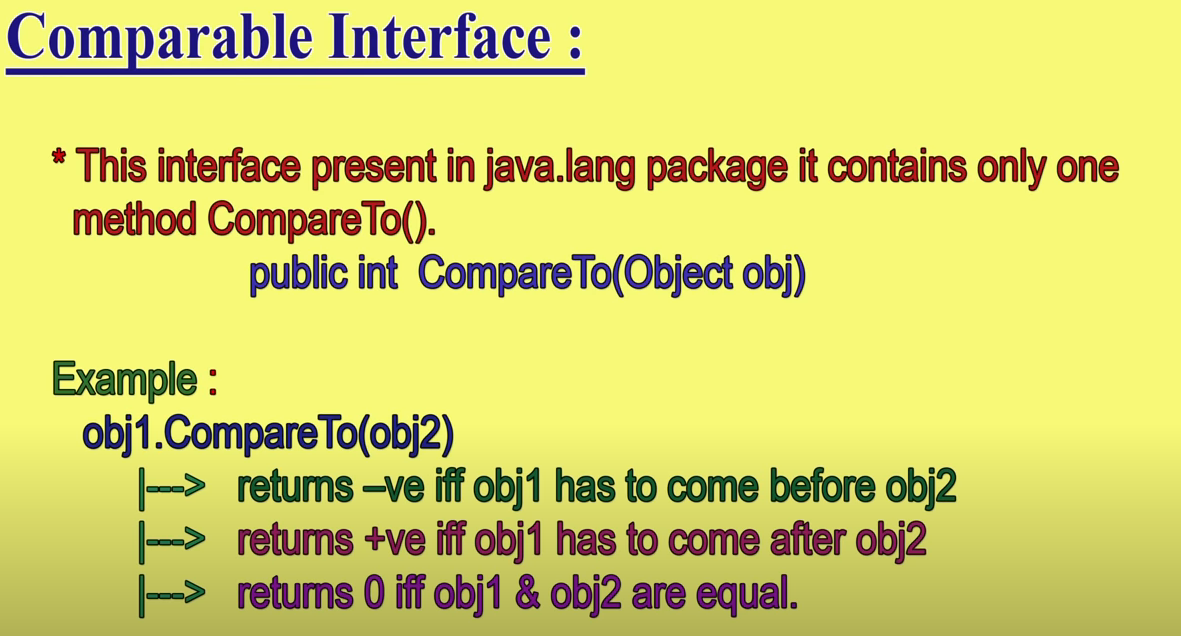


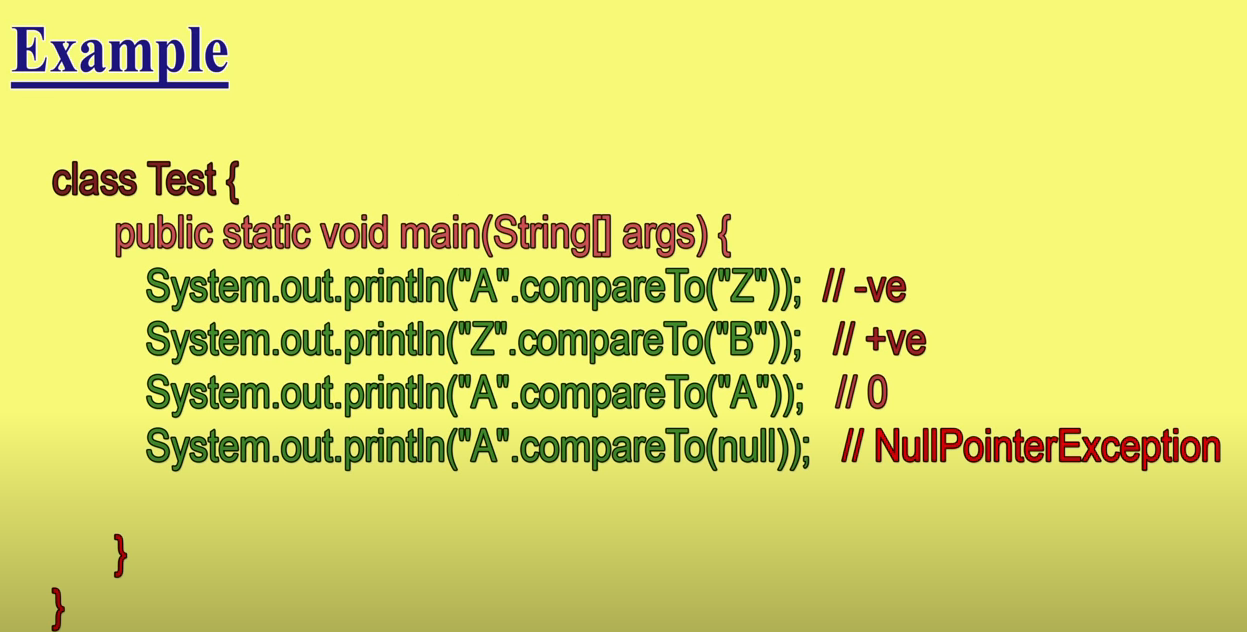


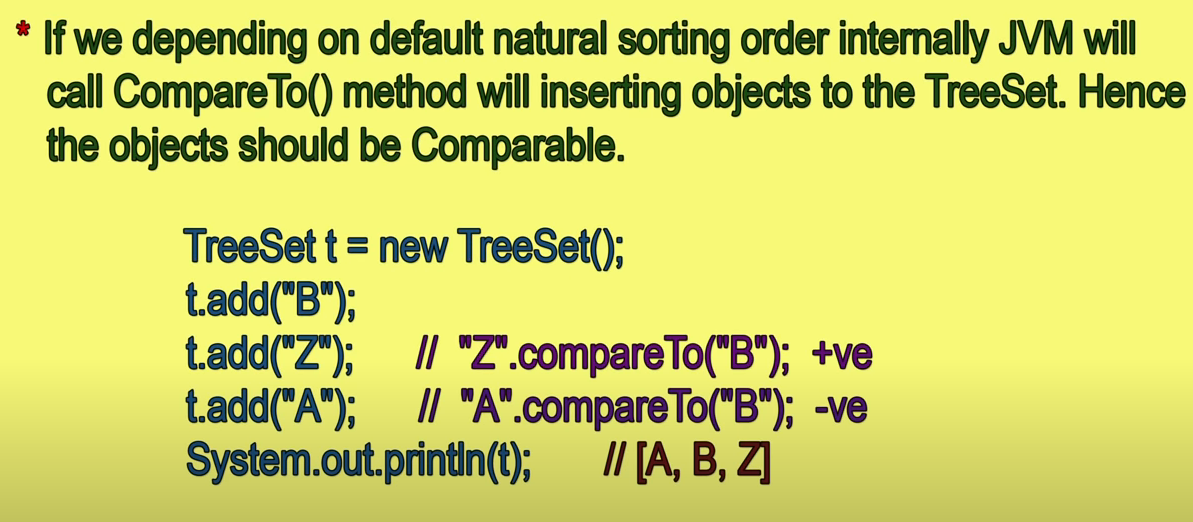


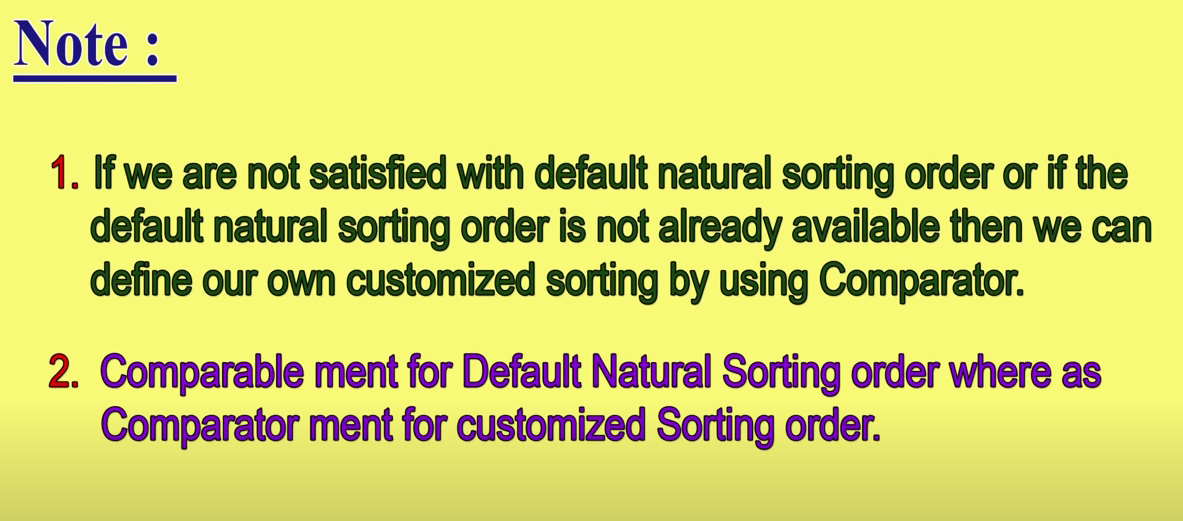


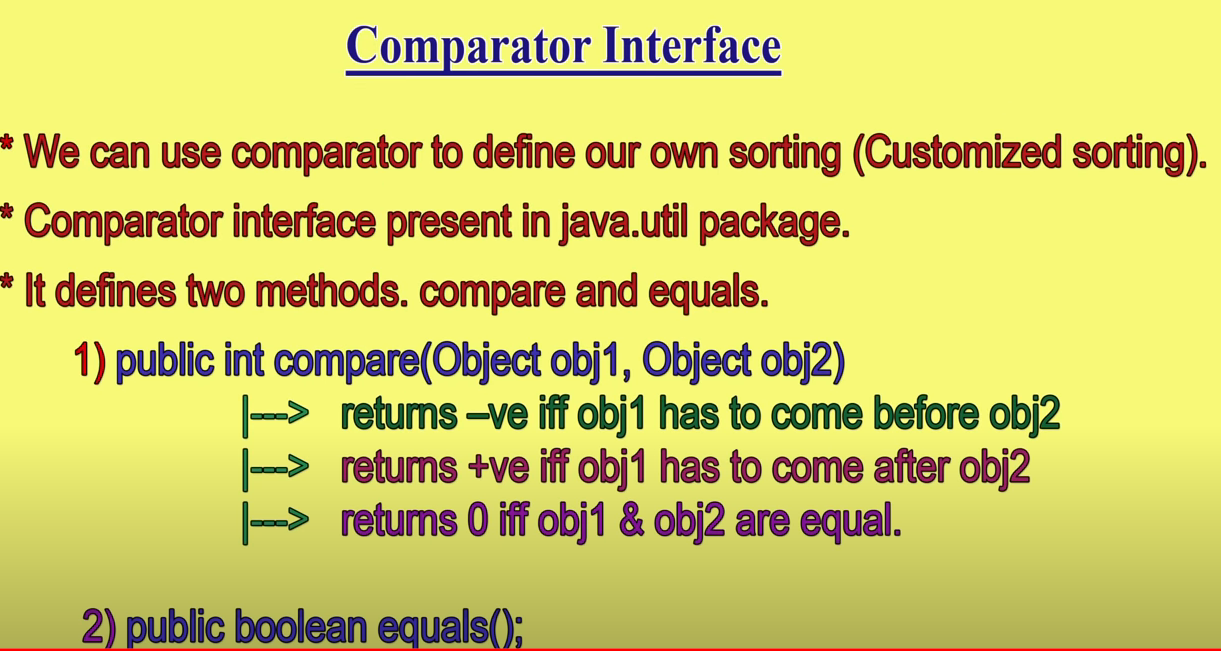


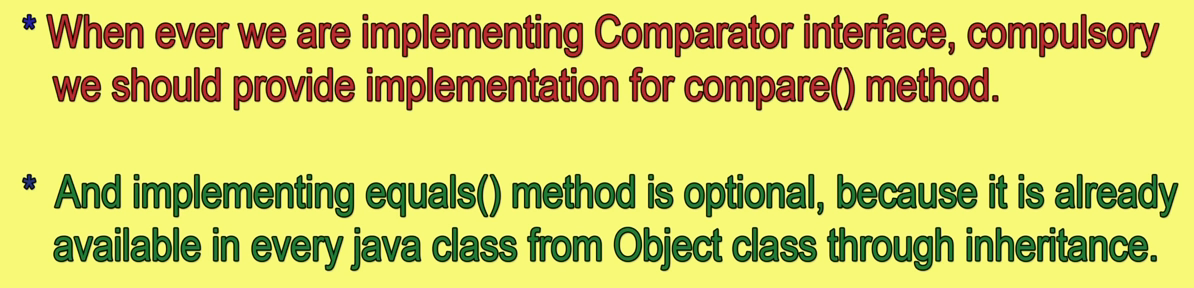


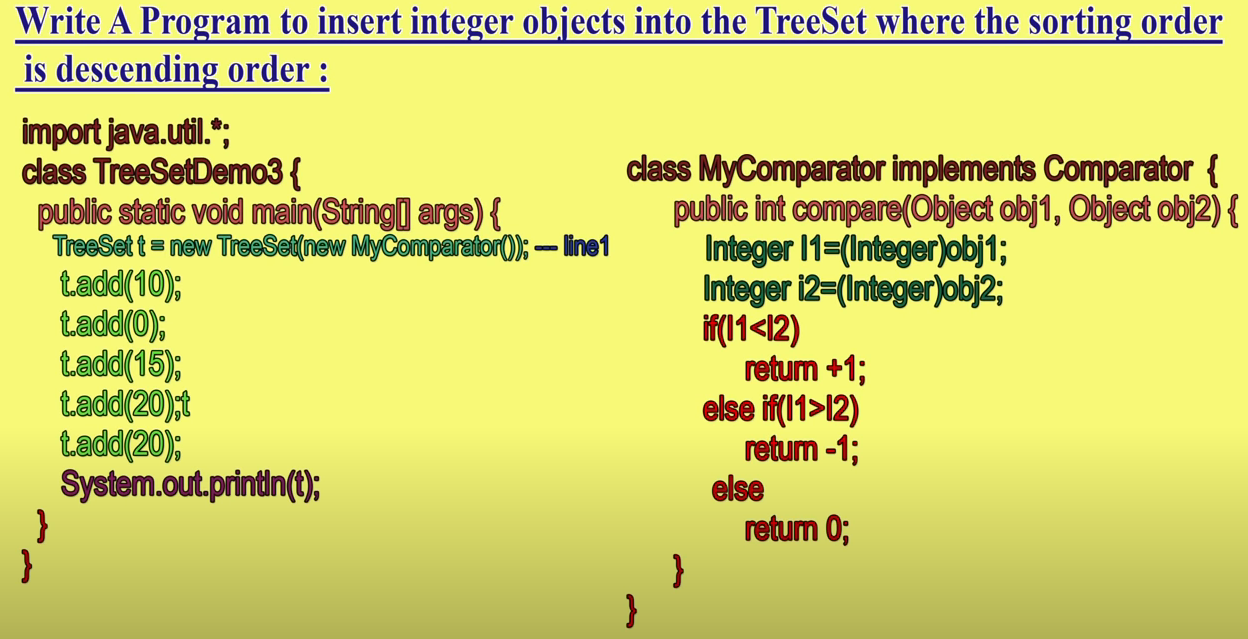


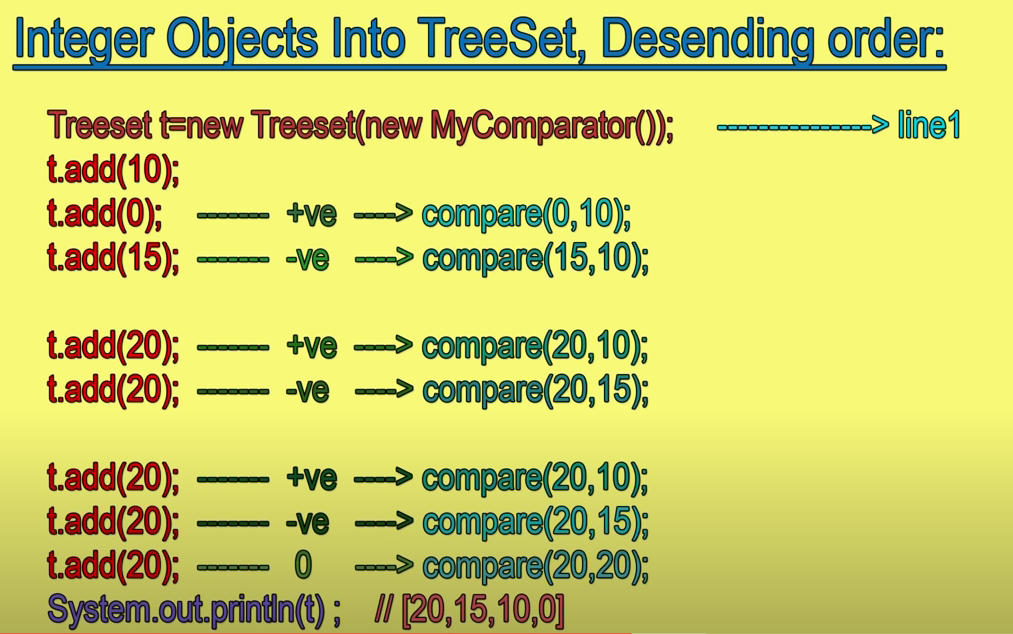


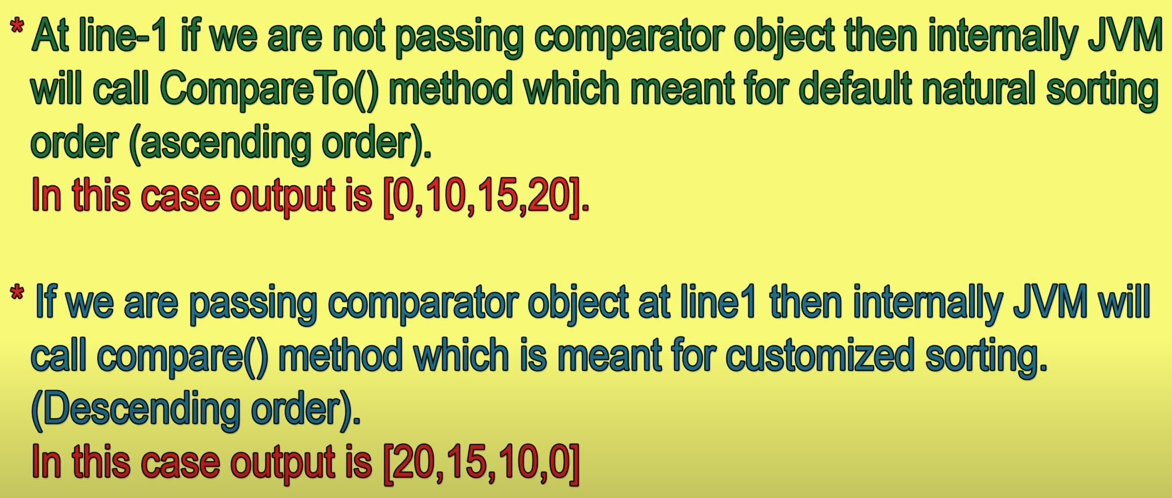


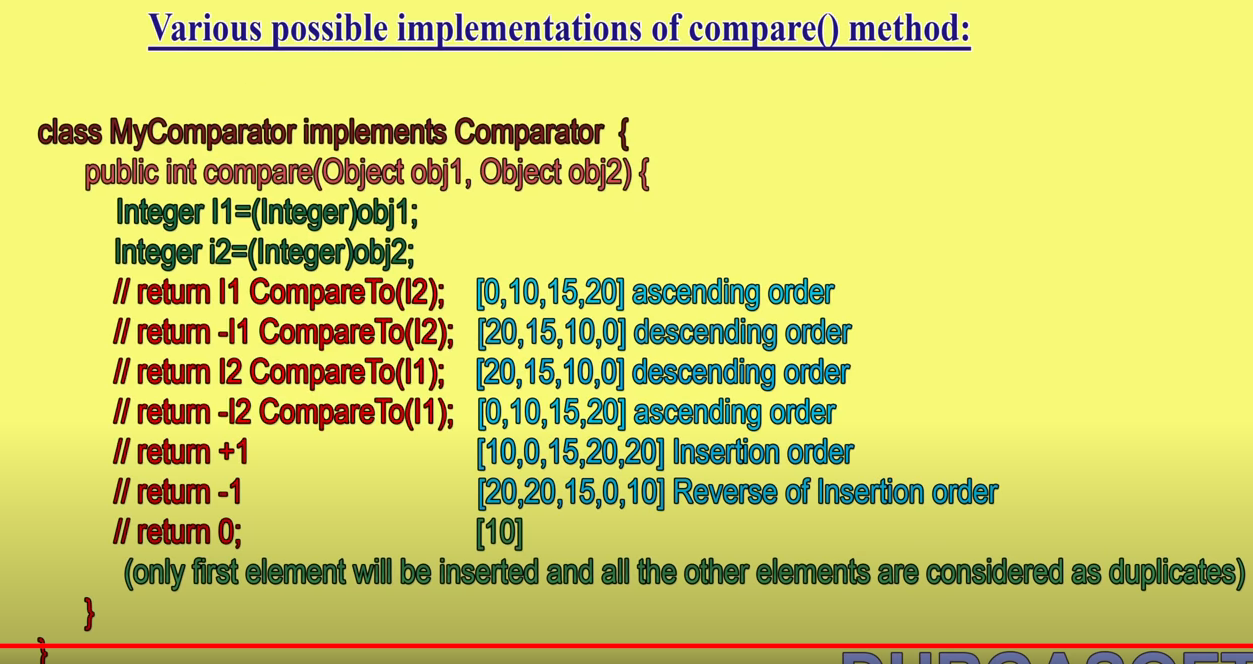












Difference between Arraylist and LinkedList

|  |  |
| --- | --- |
| **ArrayList** | **LinkedList** |
| 1) ArrayList internally uses a **dynamic array** to store the elements. | LinkedList internally uses a **doubly linked list** to store the elements. |
| 2) Manipulation with ArrayList is **slow** because it internally uses an array. If any element is removed from the array, all the bits are shifted in memory. | Manipulation with LinkedList is **faster** than ArrayList because it uses a doubly linked list, so no bit shifting is required in memory. |
| 3) An ArrayList class can **act as a list** only because it implements List only. | LinkedList class can **act as a list and queue** both because it implements List and Deque interfaces. |
| 4) ArrayList is **better for storing and accessing** data. | LinkedList is **better for manipulating** data. |