

Agenda:

- The collections framework
- Comparable and Comparator

collections → a store of items

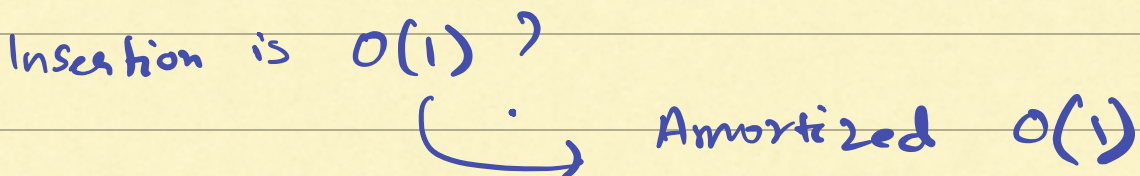
A collection is containing of interface, classes and algorithms to provide implementation of common Data Structures.

Collection Framework Hierarchy in Java

```
graph BT; Iterable[Iterable] --> Collection[Collection]; Collection --> List[List]; Collection --> Queue[Queue]; Collection --> Set[Set]; Map[Map] --> SortedMap[SortedMap]; SortedMap --> TreeMap[TreeMap]; List --> ArrayList[ArrayList]; List --> LinkedList[LinkedList]; List --> Vector[Vector]; Stack[Stack] --> Vector; Queue --> Deque[Deque]; Deque --> ArrayDeque[ArrayDeque]; Queue --> PriorityQueue[PriorityQueue]; Set --> HashSet[HashSet]; Set --> LinkedHashSet[LinkedHashSet]; Set --> SortedSet[SortedSet]; SortedSet --> TreeSet[TreeSet]; Map --> Hashtable[Hashtable]; Map --> LinkedHashMap[LinkedHashMap]; Map --> HashMap[HashMap];
```

Legend:

- Interface (Green box)
- Class (Orange box)
- Implements (Dashed arrow)
- extends (Solid arrow)



HashMap & HashSet

key values

"Akarsh"	10
"Sam"	7

hm.add("Akarsh", 10);

Unique / no duplicates are allowed.

"Akarsh"	new Object
"Sam"	new Object

HashSet
contains

HashMap's containsKey

add(E e)

↓

hm.add(e, new Object())

print

hm.keySet()

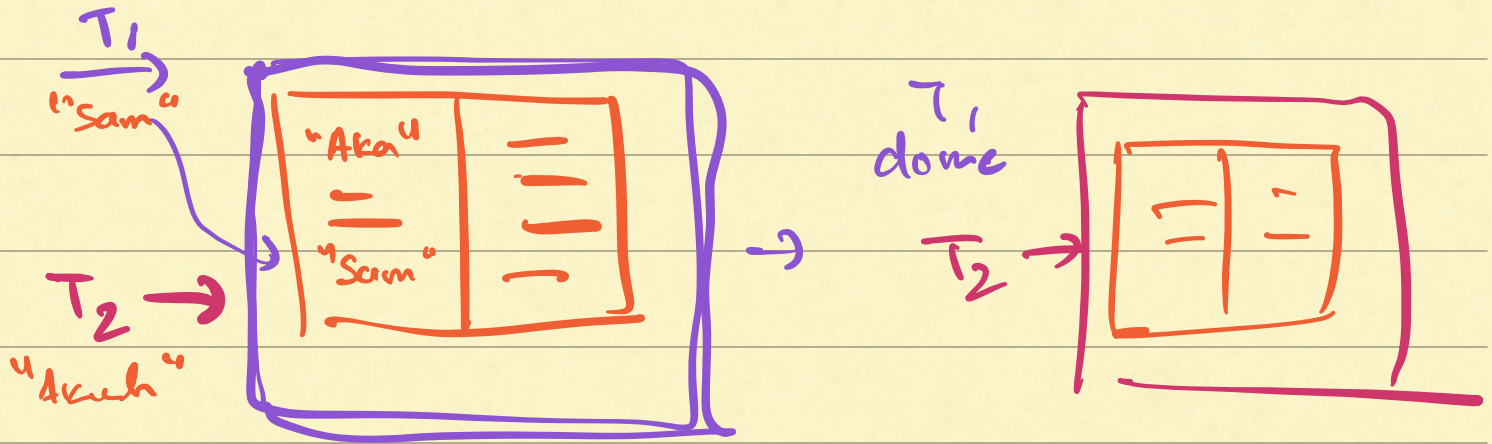
HashTable

vs

Hash Map

- Synchronised.

- Not synchronised



Concurrent Hash Map ?

↳ segment locking.

