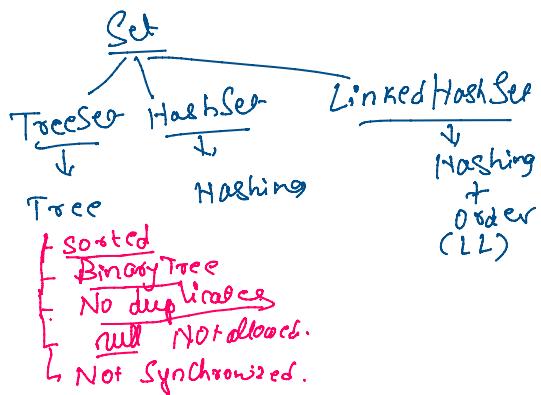
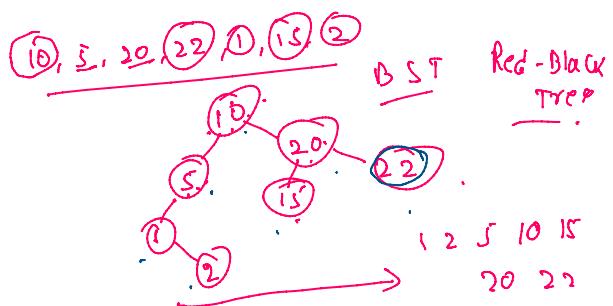


## Java Collections

Set: → doesn't allow duplicates  
→ Order may not be maintained.



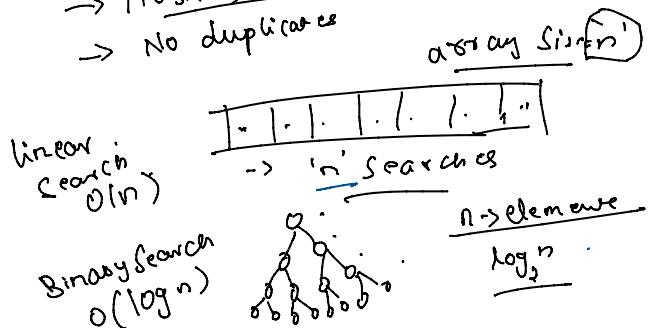
TreeSet<Integer> myTreeSet =  
new TreeSet<>();



String, double / float

## HashSet

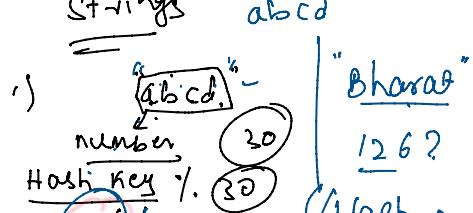
- Hashing →
- No duplicates

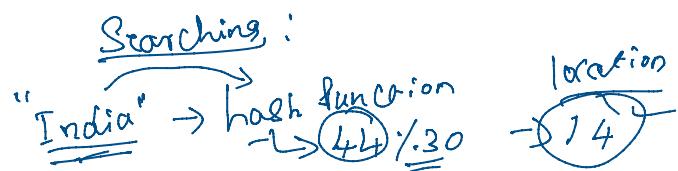
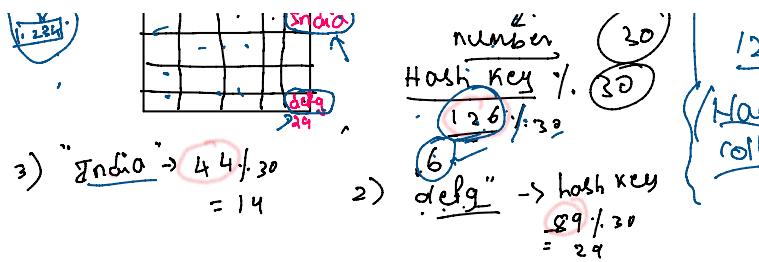


Hashing → 'i' shot  $O(1) = O(1)$



Hash Table
aabb
ccdd
eeff
gggh
iiij





HashSet  $\rightarrow$  One null permitted.

HashSet<String> hashSet = new HashSet<>(s);

hashSet.add("abcd");  
("efgh");  
("India");

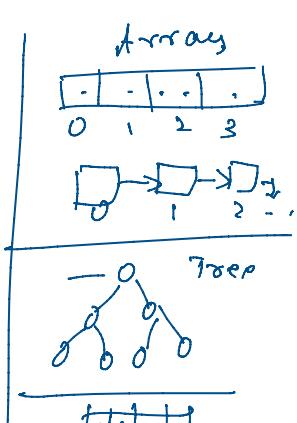
cop(hashSet);

Linked HashSet :

$\rightarrow$  Hash Set wins insertion order maintained.

Map :

Key - Value



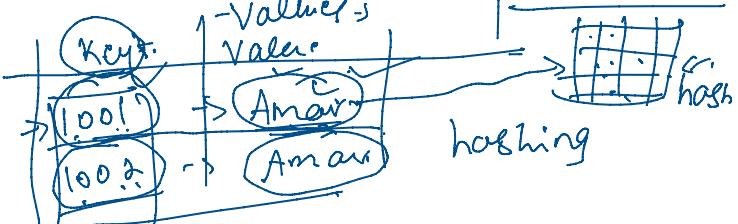
map - key  $\rightarrow$  unique

1 - Value  $\rightarrow$

Key: 1001  $\rightarrow$  Value: Amar

Key: 1002  $\rightarrow$  Value: Amar

hashing

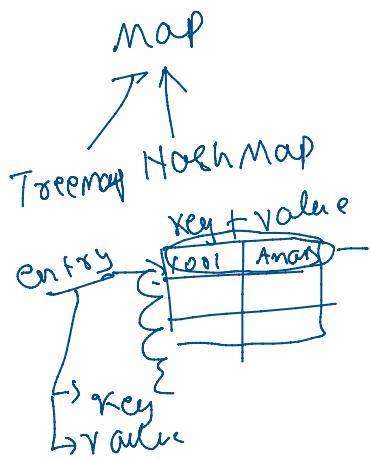
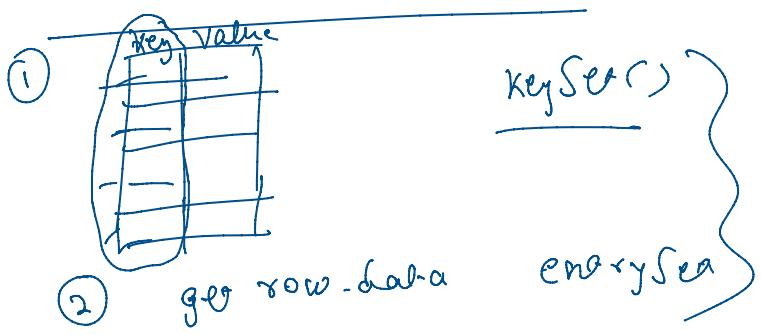


pre

Tree Map

HashMap

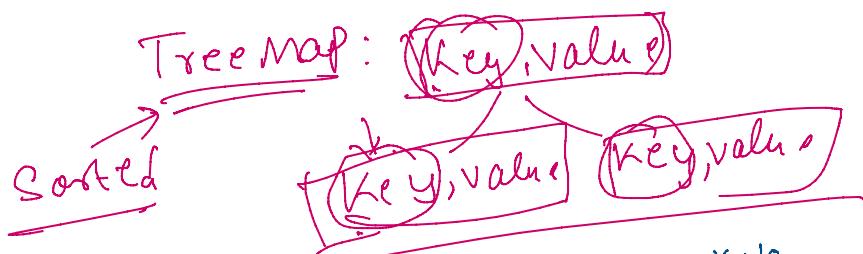
# print the contents of HashMap  
One-by-one.



- \* Create a hashmap using EmployeeId as key & EmployeeObj as value.  
Do CRUD operations.
- \* Can we use Employee Obj as key?

Employer

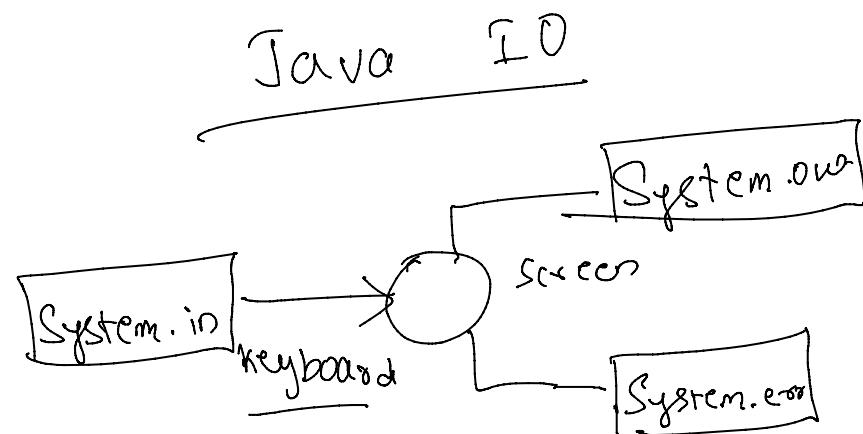
e1 = new EmpObj  
(...);



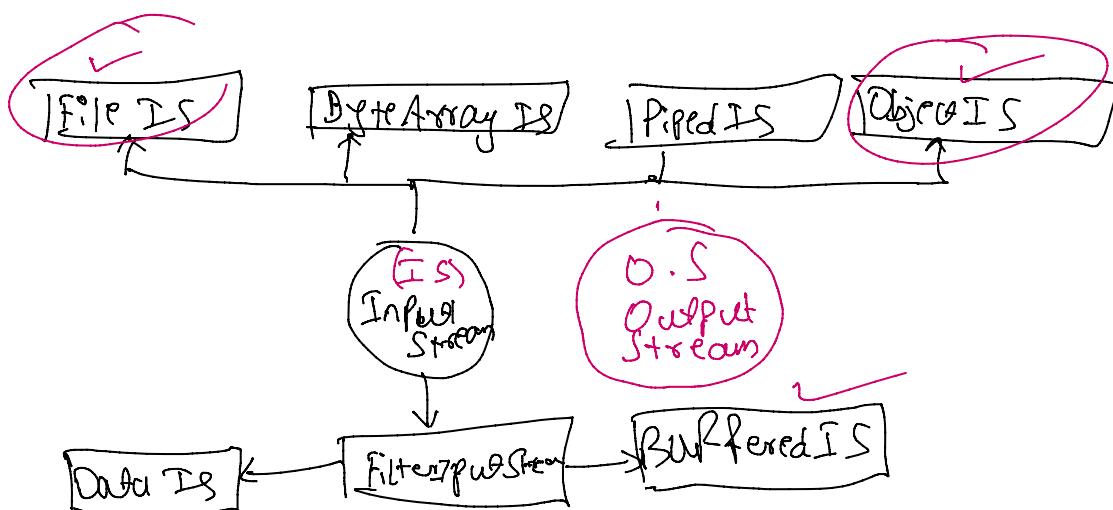
- \* Create Car Object with chassisNo, carName, fuelType, power.  
→ Insert 5 car objects in ArrayList  
→ display.  
→ Sort the car based on power (without calling ArrayList.sort())  
→ Store car details with unique names.

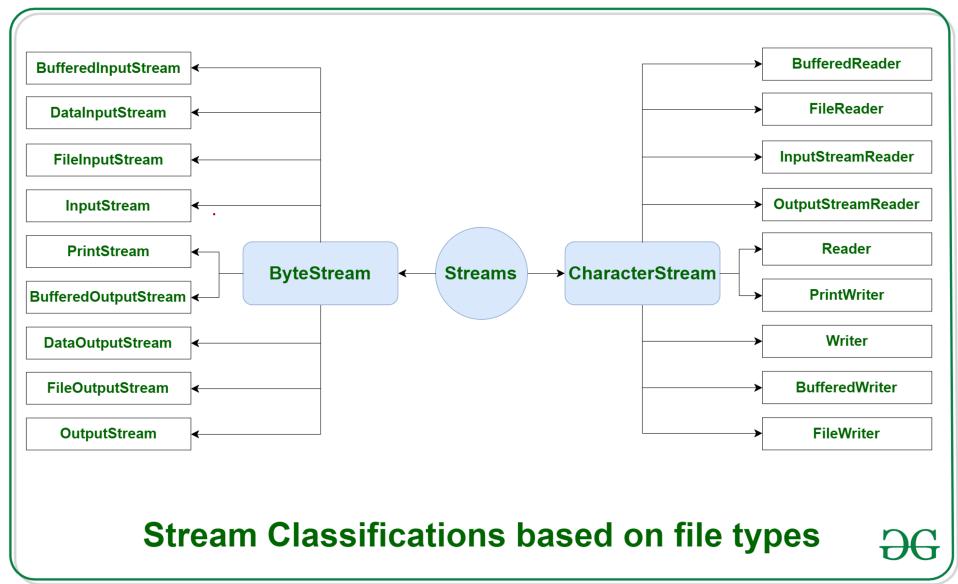
-1001	"Swift"
-1002	"Swift"
1005	"Nexon"
1007	"Dzire"

→ Display car details with unique names.



Input Streams | Output Streams.





[IIOException] ✓ all I/O related Exception  
 ↑  
 FileNotFoundException --