HashMap, HashSet- 2

Strings-3

Interfaces-4

Comparator-5

Exception-6

Arraylist, set, List-7

Polymorphism-8

Miscellaneous-At last

**HashMap and HashSet**

HashMap<I,I>- contiansKey, contiansValue, isEmpty, keySet- gives Set<>, size, put, get

Split-method-String Method

HashSet-add, remove,contains

Iterator<String> i = h.iterator();

Where h is the name of HashSet

Map.entrySet=return hashmap in list form

Map.Entry-interface for hashmap provides certain function equals <https://www.geeksforgeeks.org/map-entry-interface-java-example/>

Sorting in hashmap using above tech and comparator

Strings

String(char array) return string

**Integer.toString**- to concat int with String.

Interfaces

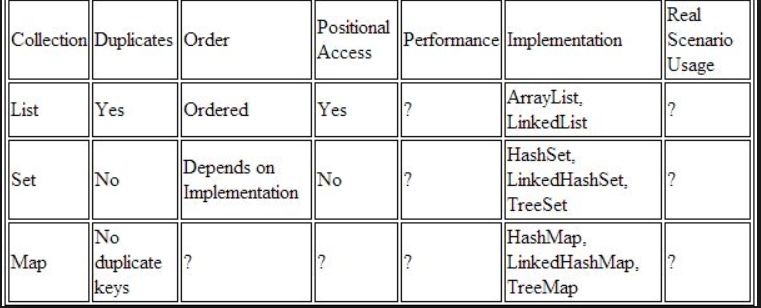
**Set** is implemented by HashSet, LinkedHashSet or TreeSet and extends collection

List- also by vector and stack- <https://www.geeksforgeeks.org/list-interface-java-examples/>

Set-<https://www.geeksforgeeks.org/set-in-java/> .All set operation like union, sub etc.

SortedSet-Interface, extends Set <https://www.geeksforgeeks.org/sortedset-java-examples/>

List vs Set both extends collection interface



Comparator

Use to sort objects

Implements comparator generic interface.

compare function and equals(one argument) function, public methods

-1 when a<b

1 when a>b sort happens

0 a==b

Then Collections.sort(collection, comparator object)/Arrays.sort()

**Comparable**-compareTo()-we can use only once, so we can only do for a particular attribute/members of class. Overriding interface function.Implemented by item class.

[comparable vs comparator](https://www.geeksforgeeks.org/comparable-vs-comparator-in-java/)

# Exception

Util package

And for creation an expection one must extend **Exception** class

ArrayList

Subtype of list

<https://www.geeksforgeeks.org/arraylist-in-java/>

add(it is used to add to specific location), by shifting all by one location

List vs ArrayList-interface and class

Interface implementation support change

**Polymorphism**

Virtual-overridden

This is known as [Polymorphism](http://java67.blogspot.com/2012/10/difference-between-polymorphism-overloading-overriding-java.html) because any virtual method will be executed from subclass only, even though they were called from super type.

All non-static methods are virtual except final ones. Overridden methods are virtual methods.

Base wala override kr deta hai. (runtime polymorphism)

Dynamic Method Dispatch-Polymorphism overriding but at runtime.

If virtual method in derived class(subclass)is private, it will raise an error.

Or assess modifier of derived is equal or more than of base.

**Collections Framework**

Collections of classes or data structures.

Collection-interface, A Collection represents a single unit of objects, i.e., a group

Collections-utility class.<https://docs.oracle.com/javase/6/docs/api/java/util/Collections.html>

**Miscellaneous**

Java heap size-<http://net-informations.com/java/cjava/limit.htm>