**Collections Framework**

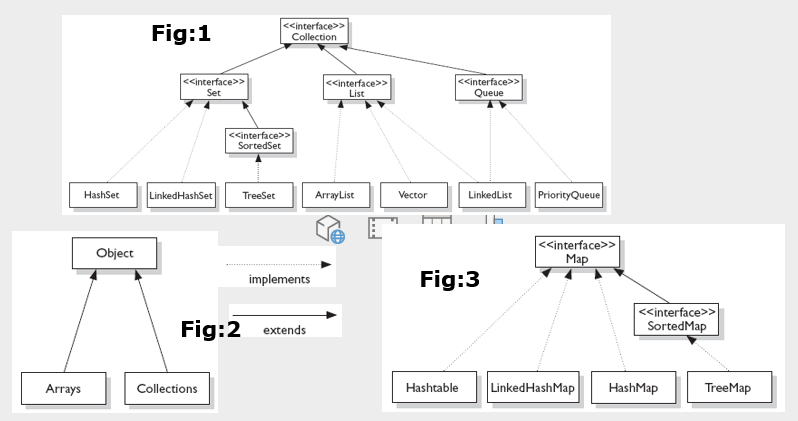
A Collection is a group of objects.

Collections framework provides a set of standard utility classes to manage collections.

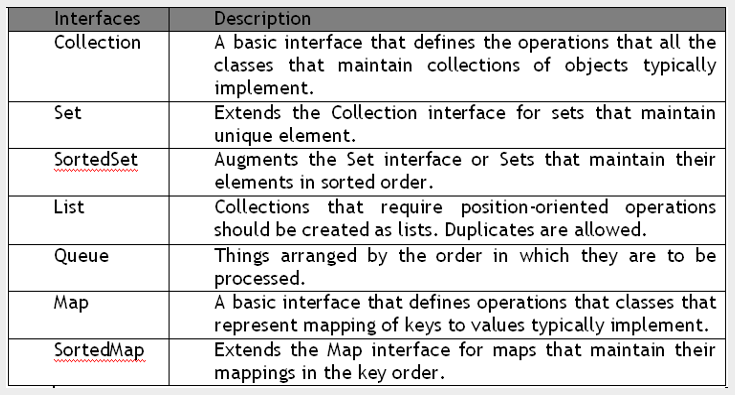
Collections Framework consists of three parts:

* + Core Interfaces
  + Concrete Implementation
  + Algorithms such as searching and sorting

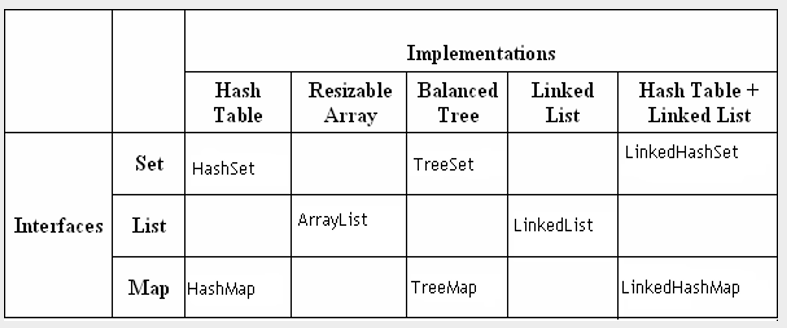
**Concept of Interfaces and Implementation**



**Collection Interfaces**



**Collection Implementations**



All collections are dynamic. ( no need to specify the size at time of creation)

All collections objects are container objects. (collection object contains reference to other objects)

The collection api (classes and interfaces) are defined inside java.util package

**List :**

* position-oriented and duplicates are allowed

ArrayList:

* + internally using array
  + index based
  + duplicated are allowed

LinkedList:

* internally using double linked list
* position based
* duplicated are allowe

**Set:**

* elements are not position based and duplicated are not allowed

**HashSet :**

* unordered and unsorted
* duplicates are not allowed

**LinkedHashSet :**

* ordered collection which maintains insertion order
* duplicates are not allowed

iterating collections:

1. using Iterator interface
2. using enhanced for loop

Type safe collections:

* collection object which accept only one type of data
* with generic we can create type safe collection

Storing user defined objects inside collection:

Lambda expression and Stream API :

What is functional interface?

* An interface contains only one abstract method

Interface MyInterface {

void test();

}

Lambda expression syntax:

(arg list) -> { expression }

Java 8 added 40 predefined functional interfaces.

Types :

1. Consumer : it takes an argument and returns nothing
2. Predicate : accepts an argument and returns a boolean
3. Supplier : it takes no arguments and returns a result.
4. Function :  it takes an argument (object of type T) and returns an object (object of type R). The argument and output can be a different type.

===============

Stream api:

* Steam objects to perform some operation on source of data ( can be array, collection etc)
* Stream operations are two types:

1. Intermediate operations

e.g) filter, map

1. Terminal operations

e.g ) forEach, count, collect