**Collection Hierarchy(interfaces)**



**

**Benefits**

* **Reduced Development Effort**
* **Better Quality**
* High-performance implementations of useful data structures and algorithms that increases the performance
* Collection is resizable and can grow.

java.util.Collection is the root interface of Collections Framework. It is on the top of the Collections framework hierarchy. It contains some important methods such as size(), iterator(), add(), remove(), clear() that every Collection class must implement.

**Set** — a collection that cannot contain duplicate elements. This interface models the mathematical set abstraction and is used to represent sets, such as the cards comprising a poker hand, the courses making up a student's schedule, or the processes running on a machine. See also The Set Interface section.

**List** — an ordered collection (sometimes called a *sequence*). Lists can contain duplicate elements. The user of a List generally has precise control over where in the list each element is inserted and can access elements by their integer index (position).

**Queue** — a collection used to hold multiple elements prior to processing. Besides basic Collection operations, a Queue provides additional insertion, extraction, and inspection operations.

**Map(separate hierarchy)-** an object that maps keys to values. A map cannot contain duplicate keys: Each key can map to at most one value.The Map interface includes methods for basic operations (such as put, get, remove, containsKey, containsValue, size, and empty),

**. 1. Diff between arraylist and linked list**

**2. Diff between hashmap and hash table**

**Implement list of students and do some basic operation ( read and write)**

**Ref Link:** [**https://docs.oracle.com/javase/tutorial/collections/interfaces/index.html**](https://docs.oracle.com/javase/tutorial/collections/interfaces/index.html)

**Source- https://github.com/samik123/CyberSuccess**