**Collections and Interfaces**

Collections are like containers that groups multiple items in a single unit.Collections are used almost in every programming language and when Java arrived, it also came with few Collection classes; Vector, Stack, Hashtable, Array. Java 1.2 provided Collections Framework that is architecture to represent and manipulate Collections in java in a standard way. Java Collections Framework consists of following parts:

* **Interfaces**: Java Collections Framework interfaces provides the abstract data type to represent collection. java.util.Collection is the root interface of Collections Framework. It is on the top of Collections framework hierarchy. It contains some important methods such as size(), iterator(), add(), remove(), clear() that every Collection class must implement. Some other important interfaces are java.util.List, java.util.Set, java.util.Queue and java.util.Map. Map is the only interface that doesn’t inherits from Collection interface but it’s part of Collections framework. All the collections framework interfaces are present in java.util package.
* **Implementation Classes**: Collections in Java provides core implementation classes for collections. We can use them to create different types of collections in java program. Some important collection classes are ArrayList, LinkedList, HashMap, TreeMap, HashSet, TreeSet.These classes solve most of our programming needs but if we need some special collection class, we can extend them to create our custom collection class.

Java 1.5 came up with thread-safe collection classes that allowed to modify Collections while iterating over it, some of them are CopyOnWriteArrayList, ConcurrentHashMap, CopyOnWriteArraySet. These classes are in java.util.concurrent package. All the collection classes are present in java.util and java.util.concurrent package.