Collections

* The **Collection in Java** is a framework that provides an architecture to store and manipulate the group of objects.
* Java Collections can achieve all the operations that you perform on a data such as searching, sorting, insertion, manipulation, and deletion.

**What is a Collection?**

* Java Collection means a single unit of objects. i.e group.
* Java Collection framework provides many interfaces - Set, List, Queue, Deque and classes such as ArrayList, Vector, LinkedList, PriorityQueue, HashSet, LinkedHashSet, TreeSet.

**What is a framework in Java?**

* It provides readymade architecture.
* It represents a set of classes and interfaces.
* It is optional.

**What is Collection framework?**

* The Collection framework represents a unified architecture for storing and manipulating a group of objects.
* It has:
* Interfaces and its implementations, i.e., classes
* Algorithm

### **Hierarchy of Collection Framework**

* The **java.util** package contains all the [classes](https://www.javatpoint.com/object-and-class-in-java) and [interfaces](https://www.javatpoint.com/interface-in-java) for the Collection framework.



## **List Interface**

* List interface is the child interface of Collection interface.
* It inhibits a list type data structure in which we can store the ordered collection of objects.
* It can have duplicate values.
* List interface is implemented by the classes ArrayList, LinkedList, Vector, and Stack.

To instantiate the List interface, we must use :

* List <data-type> list1= **new** ArrayList();
* List <data-type> list2 = **new** LinkedList();
* List <data-type> list3 = **new** Vector();
* List <data-type> list4 = **new** Stack();

## **Array List**

The ArrayList class implements the List interface. It uses a dynamic array to store the duplicate element of different data types. The ArrayList class maintains the insertion order and is non-synchronized. The elements stored in the Array List class can be randomly accessed.

## **LinkedList**

LinkedList implements the Collection interface. It uses a doubly linked list internally to store the elements. It can store the duplicate elements. It maintains the insertion order and is not synchronized. In LinkedList, the manipulation is fast because no shifting is required.

## **Vector**

Vector uses a dynamic array to store the data elements. It is similar to ArrayList. However, It is synchronized and contains many methods that are not the part of Collection framework.

## **Stack**

The stack is the subclass of Vector. It implements the last-in-first-out data structure, i.e., Stack. The stack contains all of the methods of Vector class and also provides its methods like boolean push(), boolean peek(), boolean push(object o), which defines its properties.