**Collections In Java**

* The **Collection in Java** is a framework that provides an architecture to store and manipulate the group of objects.
* Java Collection framework provides **many interfaces** (**Set, List, Queue, Deque**) and **classes** (**[ArrayList , Vector,](https://www.javatpoint.com/java-arraylist)[LinkedList,](https://www.javatpoint.com/java-linkedlist)[PriorityQueue, HashSet, LinkedHashSet, TreeSet).](https://www.javatpoint.com/java-priorityqueue)**



**Iterable Interface**

* The Iterable interface is the root interface for all the collection classes. The Collection interface extends the Iterable interface and therefore all the subclasses of Collection interface also implement the Iterable interface.
* It contains only one abstract method. i.e., => Iterator<T> iterator()

## Collection Interface

* The Collection interface is the interface which is implemented by all the classes in the collection framework. It declares the methods that every collection will have.
* Collection interface builds the foundation on which the collection framework depends.
* Some of the methods of Collection interface are Boolean add ( Object obj), Boolean addAll ( Collection c), void clear(), etc. which are implemented by all the subclasses of Collection interface.

## List Interface

## List interface is the child interface of Collection interface.

## Stores the ordered collection of objects.

## It can have duplicate values.

## List interface is implemented by the classes ArrayList, LinkedList, Vector, and Stack.

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

**Array List :**

1. **ArrayList is a class which implements List Interface.**
2. **Dynamic implementation of arrays is through ArrayList.**
3. **ArrayList maintains Indexing (maintains insertion order).**
4. **ArrayList allows duplicates.**
5. **ArrayList is Not Synchronized.**
6. **Allows random Access.**

**Object Of ArrayList**

ArrayList al = **new** ArrayList();

al.add(10);

al.add(120.0);

al.add('a'); **Adding Objects to the ArrayList**

al.add("amulya");

al.add(.01);

**for**(**int** i=0;i<al.size();i++) { Printing the ArrayList with

System.***out***.println(al.get(i)); the help of **for loop**

}

**No generic type for the ArrayList so any type of object can be stored.**

**Generic Example:**

**ArrayList<Integer> obj = new ArrayList<Ingeter>();**

**Generic (The type of data which the ArrayList contains)**

Methods:

1. Obj.size(); (returns the size or length of the ArrayList)
2. Obj.get(i); (return the object at that position)
3. Obj.add(something); (adding object to the ArrayList)
4. Collections.sort(obj);
5. Al1.retainAll(Al2) (common objects)
6. Al1.remove(i) or Al1.remove(“Amulya”);

Adding User defined Objects to the ArrayList:

(EveryThing in Same Package)

Employee.java (class)

**package** ArrayList;

**public** **class** Employee {

String name;

**int** age;

String Dpt;

**public** Employee(String name, **int** age,String Dpt) {

**this**.name = name;

**this**.age = age;

**this**.Dpt = Dpt;

}

}

**package** ArrayList;

**import** java.util.\*;

**public** **class** ArrayListExperiment {

**public** **static** **void** main(String[] args) {

Employee e1 = **new** Employee("Amulya",20,"cse");

Employee e2 = **new** Employee("Thanuj",25,"ece");

Employee e3 = **new** Employee("Naimish",30,"it");

ArrayList<Employee> emp1 = **new** ArrayList<Employee>();

emp1.add(e1);

emp1.add(e2);

emp1.add(e3);

Iterator<Employee> it = emp1.iterator();

**while**(it.hasNext()) {

Employee emp = it.next();

System.***out***.println(emp.name+" "+emp.age+" "+emp.Dpt);

}

}