**Generic:** General

**Collection Object:** That can contain multiple objects like we can store multiple teacher type objects in an array.

**Limitations of an array:**

1. Same data type
2. Size will be fixed of array in Java.

If we need a collection such that we can store multiple different objects and we can shrink and grow size by our own choice, **Java provides some Collection Objects:**

1. **ArrayList**
2. **Vector**
3. **HashMap**

**Array List:**

* It is similar to array but it provides **2 features:**

1. It can save multiple different types of objects.
2. Its size will grow and shrink dynamically.

* Whenever, you add a new instance, size will be increased by 1 and will automatically decrease by 1 when you remove an instance.
* ArrayList is in **java.util**
* You need to create instance of this class.

**ArrayList AL = new ArrayList();**

* To add an item in array list, you have an add function:

**AL.add(obj);**  //Return Type: void

* To remove an instance:

**AL.remove(index);** //Return Type: void

* To get an instance:

**AL.get(index);** //Return Type: Object

* All objects are being stored in **form of Object (data type)** and will be retrieved with Object type (Parent of all classes).
* If we want to see which class does the object belongs, you can use **instance of** operator.
* This array list can save multiple different type of objects but when you will compile it, Java will give you an alert that u must need to be careful while retrieving because there can be generic objects stored. **(Un-safe).**
* You have a scenario that you only want to store Student type objects, then open in **safe mode** and tell the Object type:

**ArrayList<Student> AL = new ArrayList<Student>();**

All objects will be saved of Student type. Now we don’t need to downcast here.

**AL.get(index);**  //Return Type: Student

* You can create array list by specifying type. No need of generic way.