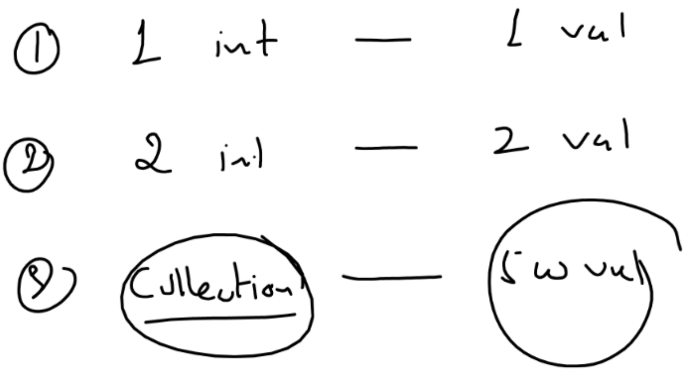
Collections



① 1 int — 1 val

② 2 int — 2 val

③ Collection — 5w val

What are Collections?
================

1. A collection in Java is a mechanism for storing group of Objects as a single entity as well as perform various operations on them.

2. These operations can be adding new object or searching an object or deleting an object or updating an object or even sorting the objects.

3. For example: Suppose we are building an E-Commerce application in Java then we can create a collection of **Product** objects, a collection of **Customer** objects, a collection of **Vendor** objects or even a collection of **Order** objects.

But before we move further with collections we might have a doubt that why don't we use Arrays and why to learn Collections?
================================
Drawbacks of Array
----
1. At the time of declaration we have to mentioned size of an Array. So following is a syntax error in Java:
int [ ] arr=new int[ ]; // Error
2. Arrays are of fixed size i.e. once created we can neither increase nor decrease size of an array.

3. Like many other popular languages an array in Java only allows HOMOGENEOUS DATA.

4. No builtin support in the form of predefined methods is available in Java for Arrays.Like there is no method in Java for adding data in an array or removing data from the array.

One Stop Solutions for all the above problem in Java is the **Collection framework.**

Benefits of Collection
================
1. Collections do not require any size to be declare at the time of declaration.

2. Collections are growable by nature i,e. at runtime they can resize themselves as the need arises.

3. Collections allow us to store HETEROGENEOUS types of data in them, although, it is not recommended.

4. Collections have very good builtin support by Java in the form of predefined methods for performing various operations on them like **add(), remove(), sort(), get(),** etc.

✓ X add(15);

Array	Collection
1. Fixed in size	1. Growable in size
2. Only allows homogeneous data.	2. Allows homogeneous as well heterogeneous data.
3. No support of predefined methods.	3. Exhaustive support of predefined methods available for collections.
4. In Java we can create Array of primitives as well as Array of objects.	4. Although, collections only allow objects to be store into them but we can store primitive in a collections because Java will automatically convert them inot wrapper class object while storing or retriving them from collections.

int	v/s	Integer
① data type		① class
② value		② object
③ int a=10;		③ ✓Integer obj = new Integer(10); or ✓ $\boxed{\text{Integer obj = 10;}}$ Autoboxing

Integr obj = 10;

---

int n;

✓ n = obj; => n = obj.intValue();

UnBoxing