**Wrapper classes in Java:**

**Wrapper classes** are used for **converting primitive data types** into **Objects.**

For example : While working with **collections in Java**, we use generics of the type safety like

**ArrayList<Integer> instead of this ArrayList<int>**. The **Integer** is a **wrapper class** of **int primitive type.**

We use wrapper class in this case because generics needs objects not primitives.

Primitive Wrapper class

byte Byte

short Short

Int Integer

long Long

float Float

double Double

char Char

Boolean Boolean

We need Wrapper is to use them in collection API. On other hand wrapper objects hold much more memory compared to primitive types.

Primitive data types are not Objects so they belong to any class. While storing in data structures which support only objects.

**Converting Primitive data type to wrapper class:**

//Converting int primitive into Integer object int num=100;

Integer obj=Integer.valueOf(num);

System.out.println(num+ " "+ obj);

**Converting Wrapper class object to Primitive data type:**

//Creating Wrapper class object

Integer obj = new Integer(100);

//Converting the wrapper object to primitive

int num = obj.intValue();

System.out.println(num+ " "+ obj); }