Wrapper Class

- The Wrapper class in java provide a mechanism to wrap the primitive into an object.
- For every primitive data type corresponding class is declared called as wrapper class.
- There are eight wrapper class declared in java. lang package which provides methods to convert primitive into an object.

Primitive data type	Wrapper classes
boolean	Boolean
char	Character
byte	Byte
short	Short
int	Integer
long	Long
float	Float
double	Double

Boxing and UnBoxing

- The process of wrapping primitive value to a corresponding Wrapper class Object is called Boxing.
- The process of converting Object to primitive value is called unboxing.

valueOf():

- We can wrap a primitive value to corresponding Wrapper class to object using valueOf().

Declaration:

public static WrapperValue valueOf(String);

public static WrapperVlaue valueOf(primitive data);

value() Method:

- This method is to find primitive value of the given wrapper Object.

Declaration:

public byte byteValue();

public short shortValue();

public int intValue();

public long longValue();

public float floatValue();

public double doubleValue();

Note:

- The six methods are implemented in all the six subclasses of a Number classes(Byte,Short,Int,Long,Double,float)
- Additional to this character class has character() and boolean() class has booleanvalue() methods.

Wrapper class Example: Primitive to Wrapper

```java
/Java program to convert primitive into objects
public class WrapperExample1{
public static void main(String args[]){
//Converting int into Integer
int a=20;
Integer i=Integer.valueOf(a);//converting int into Integer explicitly
Integer j=a;//autoboxing, now compiler will write Integer.valueOf(a) inernally
System.out.println(a+" "+i+" "+j);
}}
```

Wrapper class Example: Wrapper to Primitive

```
/Java program to convert object into primitives
//Unboxing example of Integer to int
public class WrapperExample2{
public static void main(String args[]){
//Converting Integer to int
Integer a=new Integer(3);
int i=a.intValue();//converting Integer to int explicitly
int j=a;//unboxing, now compiler will write a.intValue() int
ernally

System.out.println(a+" "+i+" "+j);
}}
```