Understanding Wrapper Classes

- In java technology if we want to represent a group of objects in the form of an object then we have to use "Collection objects", like
  - Array List
  - Vector
  - Stack
- In java applications collections objects are able to allow only group of other objects, not primitive data directly.
- If we want to store primitive data in collection objects, first we need to **convert the primitive data in object form** then we have to store, that object data in collection objects.

- Java Technology has provided the following ‘8’ number of Wrapper classes w.r.t to ‘8’ number of primitive data types.

Primitive Data Type	Wrapper Classes
byte	Byte
float	Float
short	Short
int	Integer
long	Long
double	Double
boolean	Boolean
char	Character

Points to remember…

- All most all wrapper classes define 2 constructors one can take corresponding primitive as argument and the other can take String as argument.

(except Character)

- Character class defines only one constructor which can take char primitive as argument there is no String argument constructor.
- If we are passing String as an argument in Boolean wrapper class then :

---> If the argument is true then the result also will be true irrespective of the data and case sensitiveness

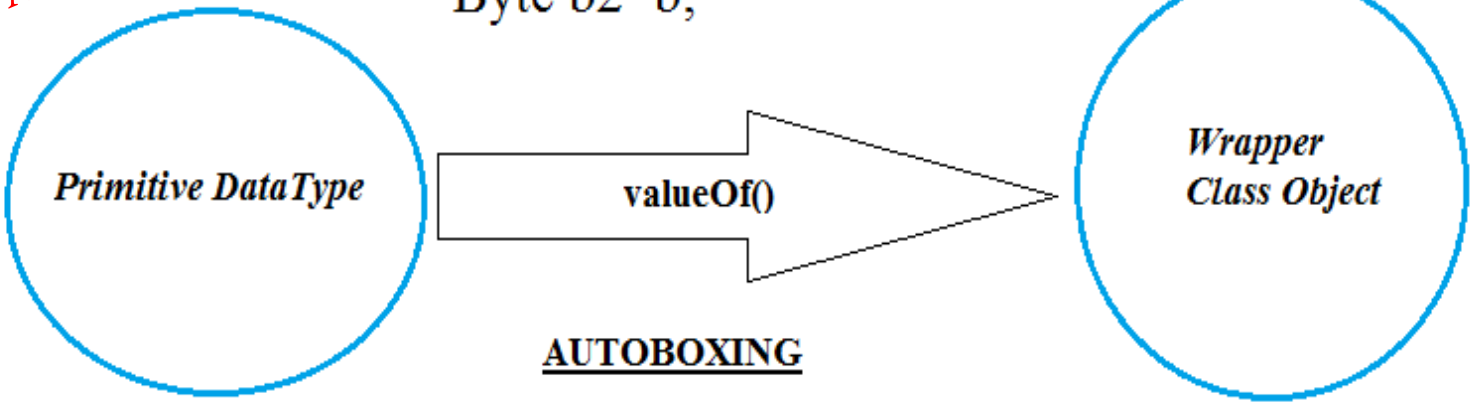
---> If the argument is false then the result also will be false irrespective of the data and case sensitiveness

---> Other than true/false any other data will give you the result as false.

COMPILER APPROACH FOR AUTO BOXING

Example for autoboxing:

```
byte b=10;
Byte b1=Byte.valueOf(b);
Byte b2=b;
```

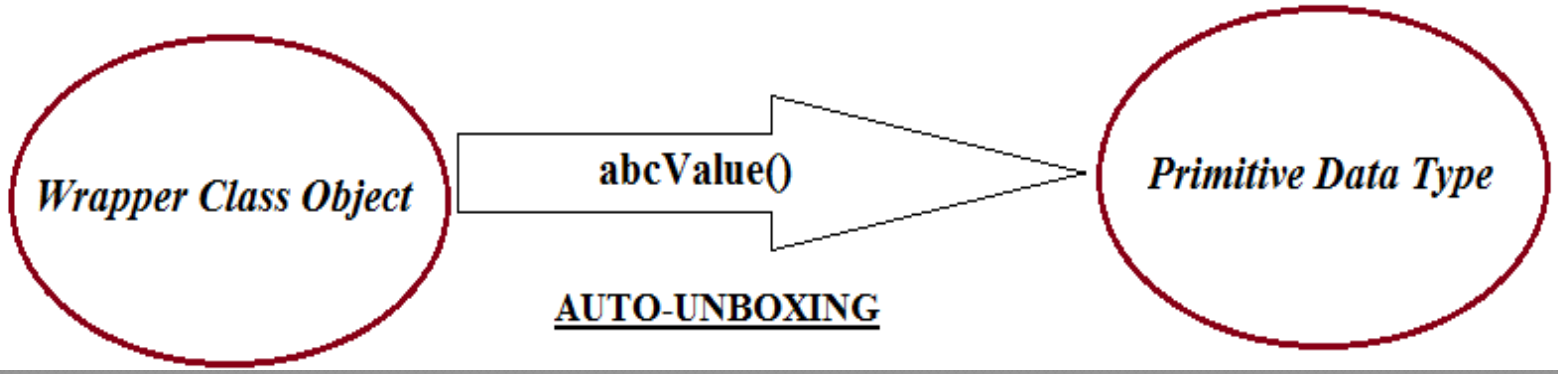


AUTOBOXING

COMPILER APPROACH FOR AUTO UN-BOXING

Example for un-boxing:

```
Byte b=new Byte("10");
byte b1=b.byteValue();
```



AUTO-UNBOXING