

# Wrapper classes

# Wrapper classes in Java

The **wrapper class in Java** provides the mechanism *to convert primitive into object and object into primitive.*

**autoboxing** and **unboxing** feature convert primitives into objects and objects into primitives automatically. The automatic conversion of primitive into an object is known as autoboxing and vice-versa unboxing.

boolean	Boolean
char	Character
byte	Byte
short	Short
int	Integer
long	Long
float	Float
double	Double

# Autoboxing

The automatic conversion of primitive data type into its corresponding wrapper class is known as autoboxing, for example, byte to Byte, char to Character, int to Integer, long to Long, float to Float, boolean to Boolean, double to Double, and short to Short.

```
//Java program to convert primitive into objects  
  
//Autoboxing example of int to Integer  
  
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) internally  
  
    System.out.println(a+" "+i+" "+j);  
  
}
```

# Unboxing

The automatic conversion of wrapper type into its corresponding primitive type is known as unboxing. It is the reverse process of autoboxing. Since Java 5, we do not need to use the intValue() method of wrapper classes to convert the wrapper type into primitives.

# Unboxing

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



*Thank you for your attention!*