Wrapper classes:

Primitive data types to object conversion and objects to primitive types conversion is taken care by wrapper classes.

1. Collections framework will works with objects so we need to convert primitive data types to object.
2. Serialization: we required objects to convert it into stream of data and process serialization.

**A** (Transmitter{Objects 🡪 stream of data[byte code] passes through network}[ **Serialization**])🡪 **B** (Receiver {stream of data[byte code] passes through network 🡪 Objects}[**deserialization**])

1. Synchronization: java multithreading synchronization will works with only objects.

2 types: **Auto boxing:**

Primitive to object conversion.

1. **Un-boxing**

Object to primitive conversion.

It was came in java 1.5v.

Primitive data type 🡺 Object data type(wrapper classes)

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

char ch = ‘k’;

Character ch = ‘k’;

Int I = 10;

Integer I = 10;

long l = 100L;

Long L = 100L;