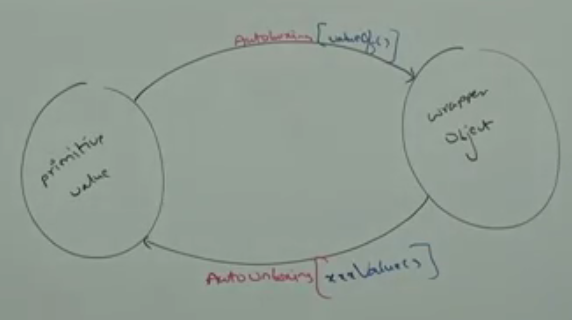
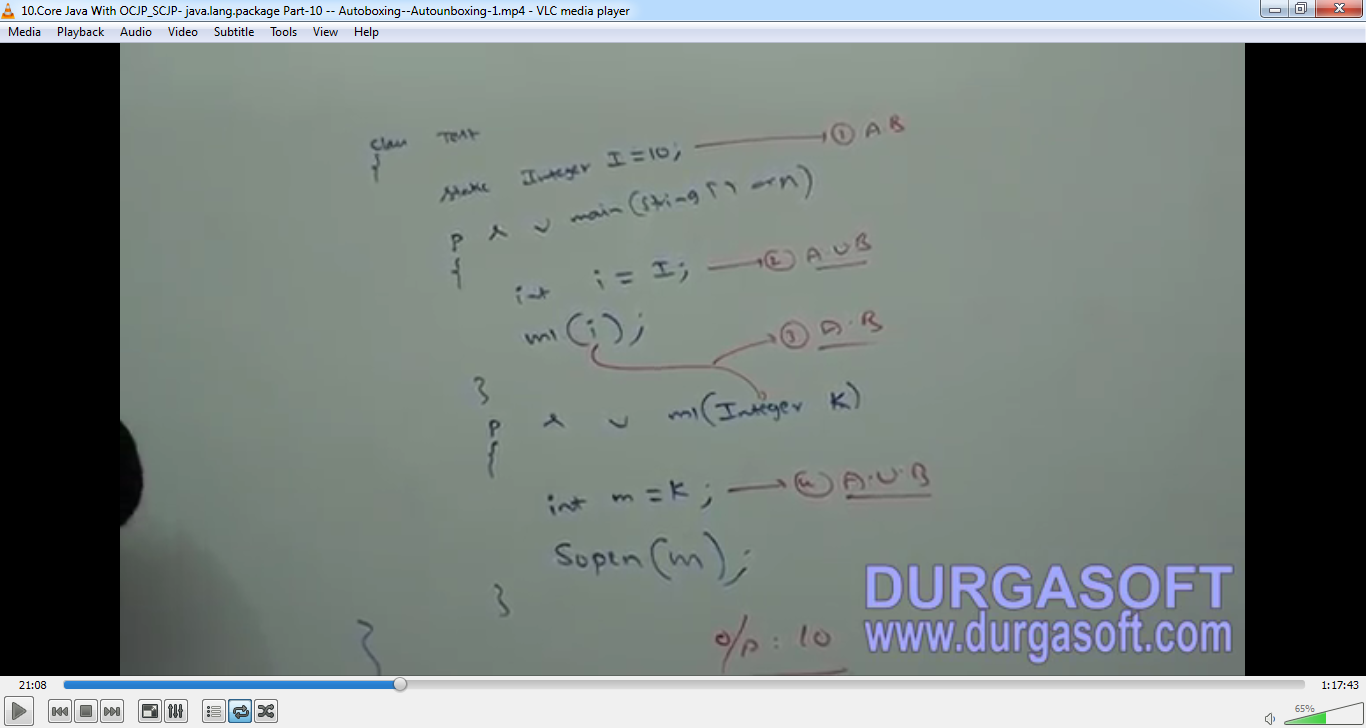
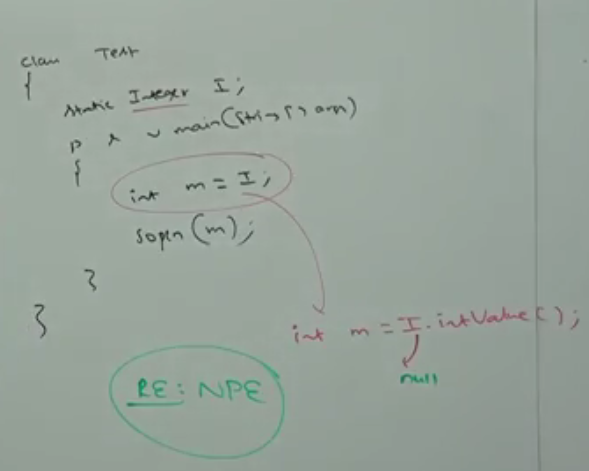
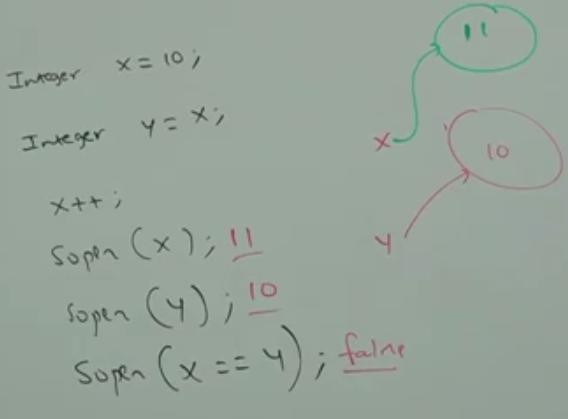
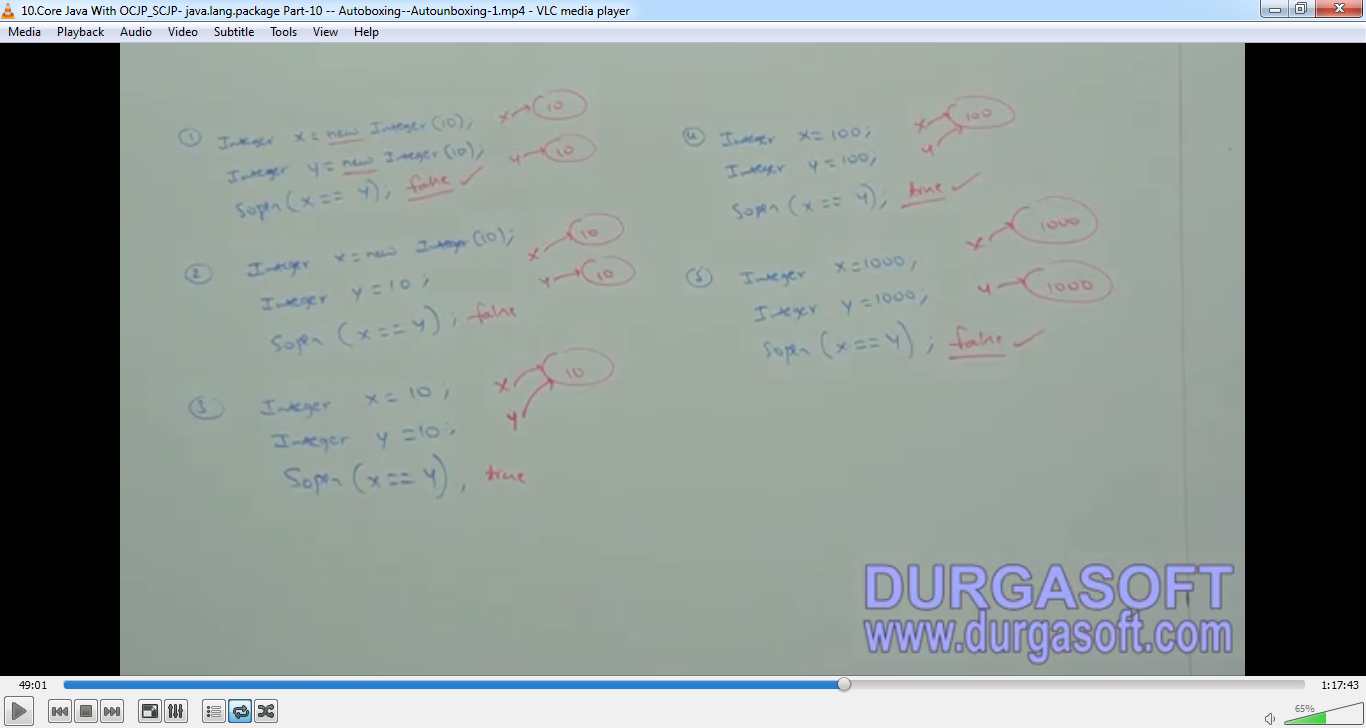
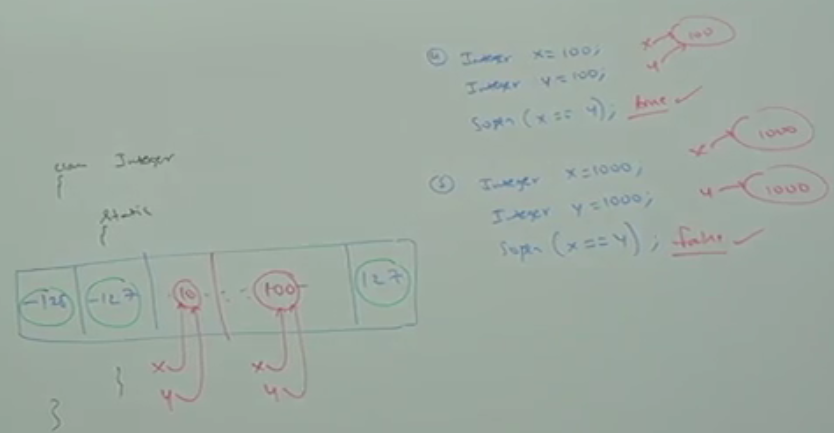
Autoboxing

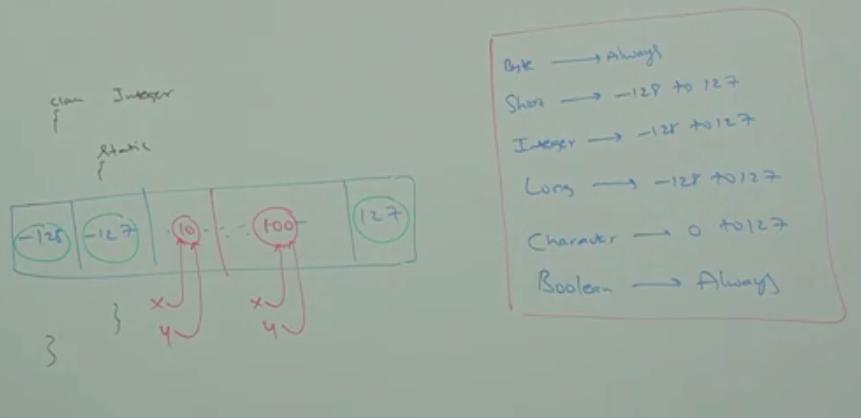
1. **AutoBoxing**:
   1. **Def**: Automatic conversion of primitive to wrapper object **by compiler** is called **Autoboxing**.
   2. **Example**:   
      Integer i = 10; // will be converted into Integer wrapper object.
   3. **Process**: After compilation, the above line will become Integer i = Integer.valueOf(10); that is internally, autoboxing concept is implemented by using valueOf() method.
2. **d**

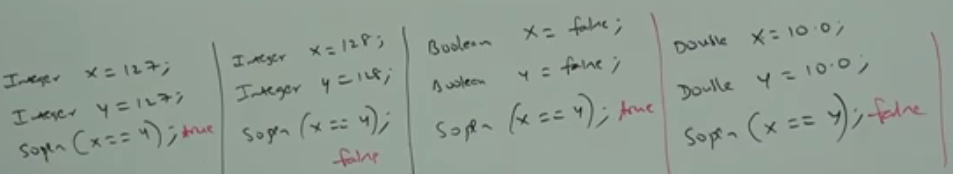
Autounboxing

1. **Autounboxing**
   1. **Def**: Automatic conversion of wrapper object to primitive by compiler is called Autounboxing
   2. **Example**:   
       Integer I = new Integer(10);  
       int i = I;
   3. **process**: After compilation, the above line (int i = I) will become int i = I.intValue();
2. 
3. 
4. NullPointerException  
   
5. **Confusing Example**:  
   

Remember: Wrapper class object is immutable

1.   
   To provide support for autoboxing, a buffer of wrapper objects will be created at the time of wrapper class loading. By autoboxing, if an object is required to create, 1st JVM will check whether this object is already present in buffer or not. If present in the buffer, the existing buffer object will be used. If not present, JVM would create a new wrapper object.  
     
     
   But buffer concept is available only in the following ranges

  
Except this range, a new wrapper object will created.

Buffer Concept is not for Float and Double you know that b/w 0-1, there are infinite numbers.  


1. Internally, autoboxing concept is implemented by using valueOf() method. Hence, buffering concept is applicable for valueOf() method also.  
   