1. The Boolean class is the object wrapper for a value of the primitive type boolean. Boolean class implements comparable interface so Boolean class needs to implement the compareTo method necessary.

**Public final class** Boolean **implements** java.io.Serializable,  
 Comparable<Boolean>  
{

...

**public int** compareTo(Boolean b) {  
 **return** *compare*(**this**.**value**, b.**value**);  
 }

**public static int** compare(**boolean** x, **boolean** y) {  
 **return** (x == y) ? 0 : (x ? 1 : -1);  
 }

...

}

That means when we are comparing two booleans through the comparison operators, true values are considered greater than False values.

2.

**List**<Integer> ints = **new** ArrayList<Integer>();  
ints.add(2);  
**List**<? **extends** Number> nums = ints;

We can assign variable ints to nums because of the use of wildcard. Type List<? extends Number> is compatible with List<Integer>. Between type List<? extends number> and List<Integer> there is a "is a relationship" and we can do the assignment operation

nums.add(3.14); *//This is not allowed now after setting an upper bound*

It is not allowed because of it is not possible to infer the type of List<? extends number>. Although the instance of the variable nums can contain a List<integer>, the type of the variable is List<? extends number> so It can be interpreted like an anything that extends of Number.

We do not add or edit elements of variable nums.