**Difference between String, StringBuffer & StringBuilder:**

String is immutable while StringBuffer is mutable means you can modify a StringBuffer object once you created it without creating any new object. This mutable property makes StringBuffer an ideal choice for dealing with Strings in Java. You can convert a StringBuffer into String by its toString() method.   
  
**StringBuffer** is very good with mutable String but it has one disadvantage all its public methods are [synchronized](http://javarevisited.blogspot.sg/2011/04/synchronization-in-java-synchronized.html)which makes it [thread-safe](http://javarevisited.blogspot.sg/2012/01/how-to-write-thread-safe-code-in-java.html) but same time slow. In JDK 5 they provided similar class called StringBuilder in Java which is a copy of StringBuffer but without synchronization.Try to use ***StringBuilder*** whenever possible it performs better in most of cases than StringBuffer class, Concatenation operator "+" is internal implemented using either StringBuffer or StringBuilder.

**Why String is Immutable of final in java:**1.Imagine String Pool facility without making string immutable , it’s not possible at all because in case of string pool one string object/literal e.g. "Test" has referenced by many [reference variables](http://javarevisited.blogspot.sg/2012/02/difference-between-instance-class-and.html) , so if any one of them change the value others will be automatically gets affected i.e. let’s say  
  
String A = "Test"  
String B = "Test"   
  
Now String B called "Test".toUpperCase() which change the same object into "TEST" , so A will also be "TEST" which is not desirable.

**System.out.println explanation**

**System** is a final class from java.lang package.  
**out** is the reference of PrintStream class and a static member of System class.  
**println** is a method of PrintStream class.