1. **How Strings are Stored in Memory? Difference b/w string literals and String object creation with new keyword.**

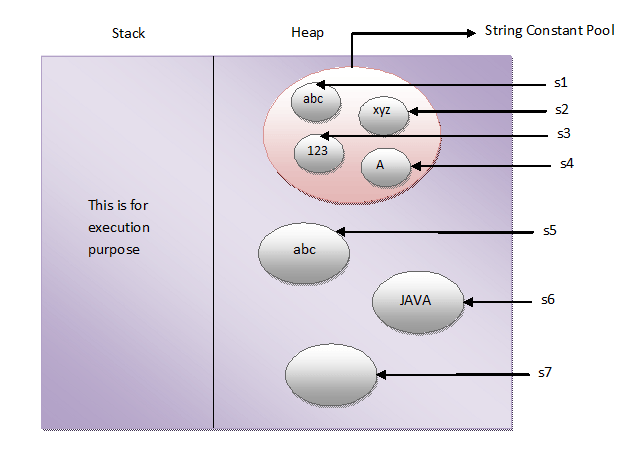
We all know that JVM divides the allocated memory to a Java program into two parts. one is **Stack** and another one is **heap**. Stack is used for execution purpose and heap is used for storage purpose. In that heap memory, JVM allocates some memory specially meant for string literals. This part of the heap memory is called **String Constant Pool**.Whenever you create a string object using string literal, that object is stored in the **string constant pool** and whenever you create a string object using new keyword, such object is stored in the heap memory.

String s1 = "abc";// Stored in Java Constant loopString

s5 = new String("abc"); //Stored in Heap memory

**When you create a string object using string literal, JVM first checks the content of to be created object. If there exist an object in the pool with the same content, then it returns the reference of that object. It doesn’t create new object. If the content is different from the existing objects then only it creates new object**

**.**But, when you create string objects using new keyword, a new object is created whether the content is same or not.



1. **What is singleton Pattern.**

[Singleton](https://www.journaldev.com/1377/java-singleton-design-pattern-best-practices-examples) pattern restricts the instantiation of a class and ensures that only one instance of the class exists in the Java virtual machine.

The singleton class must provide a global access point to get the instance of the class.

**Java Singleton Pattern Implementation**

To implement a Singleton pattern, we have different approaches but all of them have the following common concepts.

* Private constructor to restrict instantiation of the class from other classes.
* Private static variable of the same class that is the only instance of the class.
* Public static method that returns the instance of the class, this is the global access point for outer world to get the instance of the singleton class.

1. **Rules for making immutable object.**

* Don't provide "setter" methods — methods that modify fields or objects referred to by fields.
* Make all fields final and private.
* Don't allow subclasses to override methods. The simplest way to do this is to declare the class as final. A more sophisticated approach is to make the constructor private and construct instances in factory methods.
* If the instance fields include references to mutable objects, don't allow those objects to be changed:
  + Don't provide methods that modify the mutable objects.
  + Don't share references to the mutable objects. Never store references to external, mutable objects passed to the constructor; if necessary, create copies, and store references to the copies. Similarly, create copies of your internal mutable objects when necessary to avoid returning the originals in your methods.

# Java Character Set

* The java.nio.charset package provides classes to encode/decode a CharBuffer to a ByteBuffer and vice versa.
* An object of the Charset class represents the encoding scheme. The CharsetEncoder class performs the encoding. The CharsetDecoder class performs the decoding.
* We can get an object of the Charset class using its forName() method by passing the name of the character set as its argument.
* For simple encoding and decoding tasks, we can use the encode() and decode() methods of the Charset class.