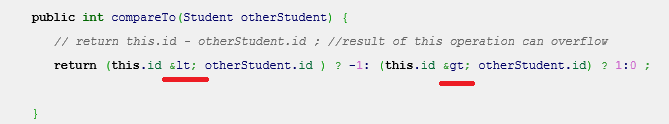
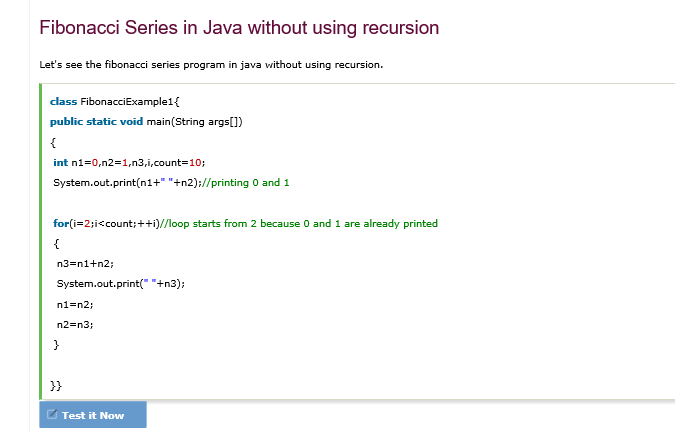
1. If you see then logical difference between these two is *Comparator in Java* compare two objects provided to him, while Comparable interface compares "this" reference with the object specified.
2. 
3. Comparable in Java is used to implement **natural ordering of object**. In Java API String, Date and wrapper classes implements Comparable interface.Its always good practice to override compareTo() for value objects.
4. If any class implement Comparable interface in Java then collection of that object either [List](http://javarevisited.blogspot.sg/2012/04/difference-between-list-and-set-in-java.html) or Array can be sorted automatically by using  Collections.sort() or Arrays.sort() method and object will be sorted based on there natural order defined by CompareTo method.  
     
   Read more: <http://javarevisited.blogspot.com/2011/06/comparator-and-comparable-in-java.html#ixzz574zJ6NTF>
5. Objects which implement *Comparable in Java*  can be used as keys in a SortedMap like [TreeMap](http://javarevisited.blogspot.sg/2011/12/treemap-java-tutorial-example-program.html) or elements in a SortedSet  for example TreeSet, without specifying any Comparator.
6. **Write a Java program to print Fibonacci series upto 100?**

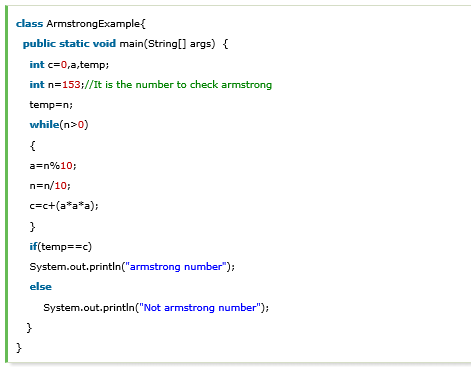


1. **Write a Java program to check if a number is Armstrong or not ?**

Armstrong Number in Java: **Armstrong number** is a number that is equal to the sum of cubes of its digits for example 0, 1, 153, 370, 371, 407 etc.

Let's try to understand why **153** is an Armstrong number.

153 = (1\*1\*1)+(5\*5\*5)+(3\*3\*3)



1. **Write a Java program to prevent deadlock in Java ?**  
   The deadlock happens if four condition is true e.g. mutual exclusion, no waiting, circular wait and no preemption. If you can break any of this condition than you can create Java programs,which are deadlock proof

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

From journal dev 🡺

### How to Detect Deadlock in Java

To detect a deadlock in java, we need to look at the [java thread dump](https://www.journaldev.com/1053/java-thread-dump-visualvm-jstack-kill-3-jcmd) of the application, in last post I explained how we can generate thread dump using VisualVM profiler or using jstack utility

Java Thread Dump – VisualVM, jstack, kill -3, jcmd 🡺 <https://www.journaldev.com/1053/java-thread-dump-visualvm-jstack-kill-3-jcmd>

Java Thread dump is list of all the threads active in the JVM.

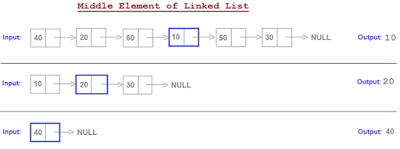
Java thread dump is very helpful in analyzing bottlenecks in the application and [deadlock](https://www.journaldev.com/1058/deadlock-in-java-example) situations

Here we will learn multiple ways through which we can generate thread dump for a java program. These instructions are valid for \*nix operating systems but in windows the steps might be little different. 🡺 ms I HAVE NOT COVERED THIS PORTION LATER WE WILL SEE FOR WINDOWS

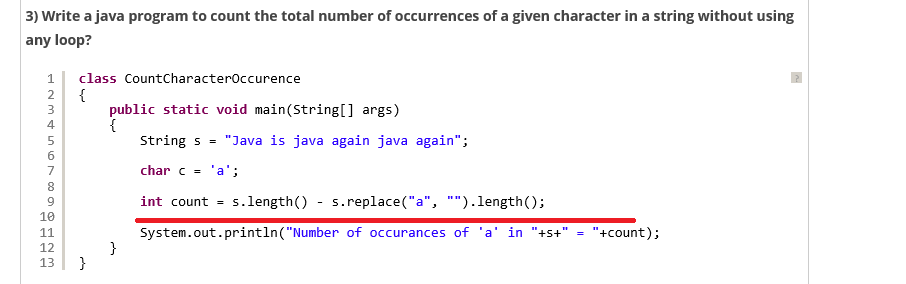
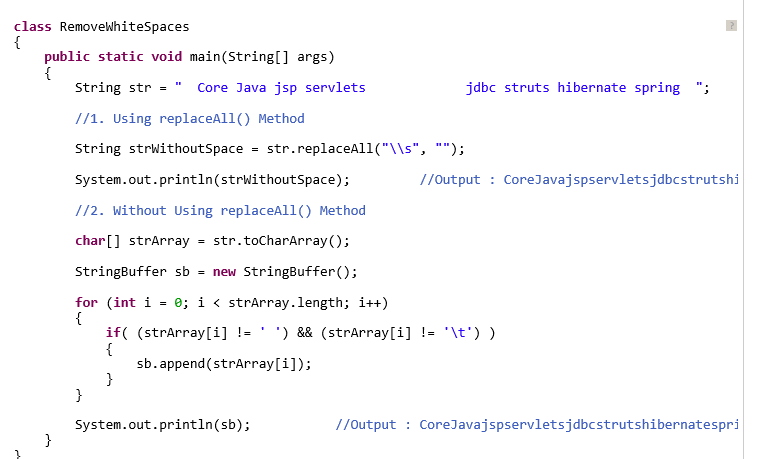
Thread dump is the list of all the threads, every entry shows information about thread which includes following in the order of appearance.

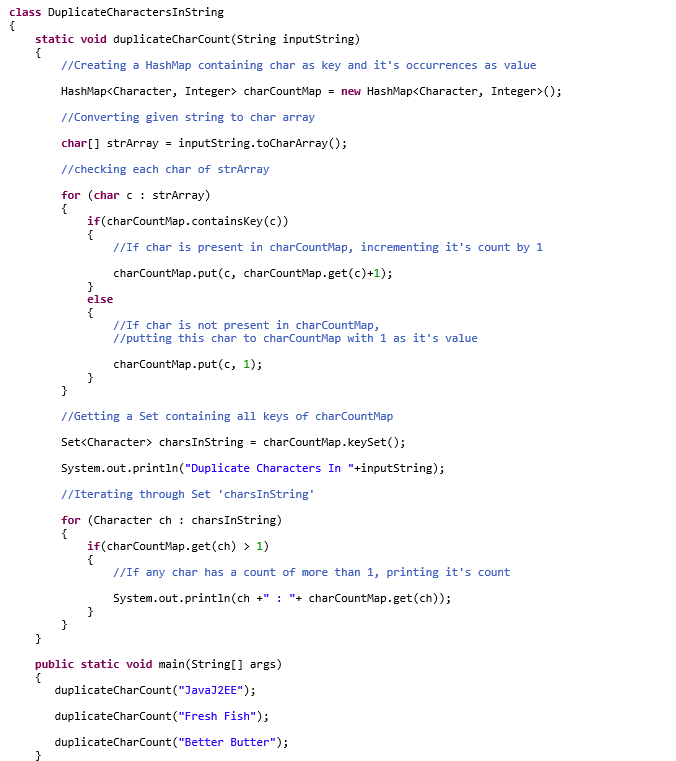
1. **Thread Name**: Name of the Thread
2. **Thread Priority**: Priority of the thread
3. **Thread ID**: Represents the unique ID of the Thread
4. **Thread Status**: Provides the current [thread state](https://www.journaldev.com/1044/thread-life-cycle-in-java-thread-states-in-java), for example RUNNABLE, WAITING, BLOCKED. While analyzing deadlock look for the blocked threads and resources on which they are trying to acquire lock.
5. **Thread callstack**: Provides the vital stack information for the thread. This is the place where we can see the locks obtained by Thread and if it’s waiting for any lock.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. Write a Java program to find if a number is prime number or not?
2. **Create a Java program to find middle node of linked list in Java in one pass?** Trick to solve this problem is to remember that last node of linked list points to null and you can trade memory with speed.  
   [](https://1.bp.blogspot.com/-x5x4IwSHtFc/WTO0Ug3wlQI/AAAAAAAAIwo/DsG-oGWwh14gBcxKim_Zlzavf8-hdzj1ACLcB/s1600/middle%2Belement%2Bof%2Blinked%2Blist%2Bin%2BJava.png)
3. **Write a Java program to calculate Factorial of a number in Java?**

<http://javaconceptoftheday.com/append-text-to-a-file-in-java/>

1. **Write a java program to find the duplicate words and their number of occurrences in a string?**
2. 
3. Write a java program to remove all white spaces from a string.?
4. **Using replaceAll() Method**
5. **Without Using replaceAll() Method**
6. [How To Find Duplicate Characters In A String In Java?](http://javaconceptoftheday.com/duplicate-characters-in-a-string-in-java/)



1. **Write a java program to check whether two strings are anagram or not?**

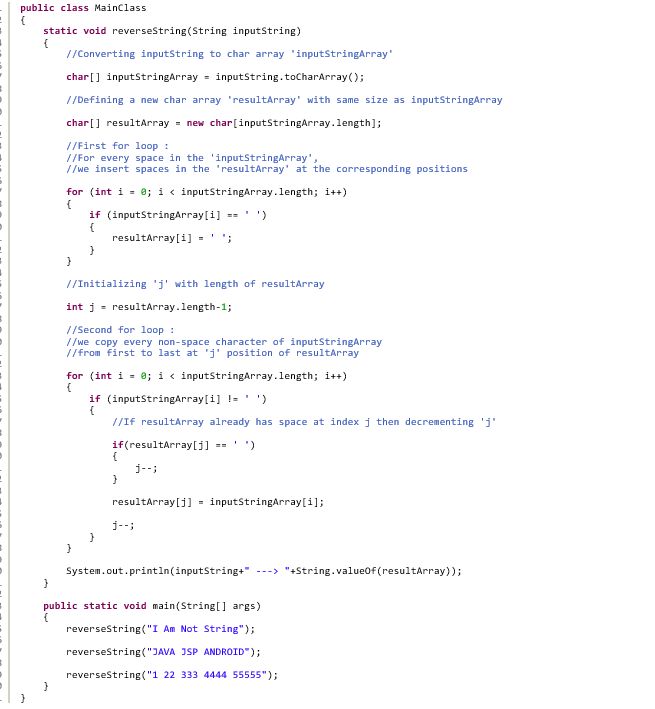
Two strings are called anagrams if they contain same set of characters but in different order. For example, **“Dormitory – Dirty Room”**, **“keep – peek”,  “School Master – The Classroom”** are some anagrams.

Logic

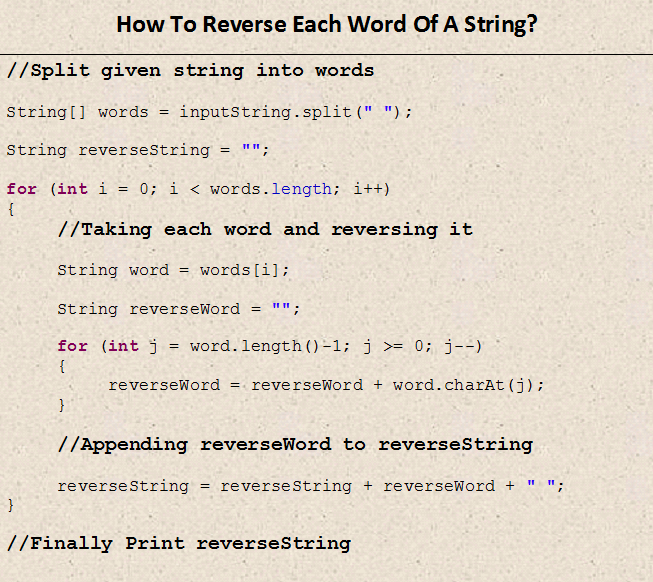
First we clean the input by removing all white spaces from the given two strings and change the case of all characters of both the strings to lower case so that case of both input strings will be ignored. After cleaning the input strings, we convert them to character array and sort them using **sort() method** of java.util.Arrays class. After sorting, we compare both the arrays using **equals() method** of same Arrays class.This method will return true if both arrays have same set of characters

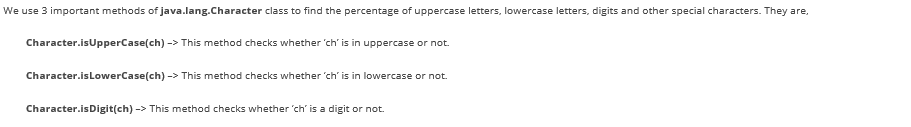
1. **Write a java program to reverse a given string with preserving the position of spaces?**

First, we convert the given **‘inputstring’** to char array and call it as **‘inputStringArray’**. We define one more char array called **‘resultArray’** with the same size as **‘inputStringArray’**. In the first for loop, for every space in the **‘inputStringArray’**, we insert space in the **‘resultArray’**at the corresponding positions. In the second for loop, we copy non-space characters of **‘inputStringArray’** starting from first to last into the **‘resultArray’** at **‘j’** position where **‘j’** will have value from **length of resultArray** to **0**. Before copying, we check whether the **‘resultArray’** already contains a space at index ‘j’ or not. If it contains, we copy the character in the next position. See the below image for more clarity.

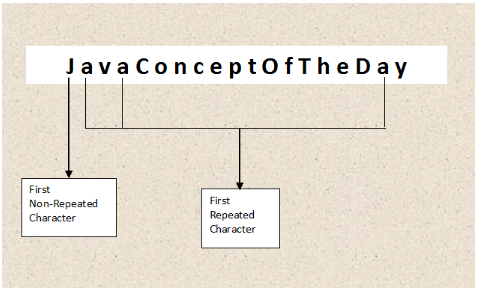
****

1. **Write a java program to reverse each word of a given string?**



1. **Write a java program to find the percentage of uppercase letters, lowercase letters, digits and special characters in a given string?**
2. **Write a java program to prove that strings are immutable in java?**
3. **How do you find longest substring without repeating characters in the given string?**
4. **How do you find first repeated and non-repeated character in the given string in java?**

Given a string, your code must find out the first repeated as well as non-repeated character in that string. For example, if “**JavaConceptOfTheDay**” is the given string, then **‘J’** is a first non-repeated character and **‘a’** is a first repeated character.  Have a look at the below image



How To Find First Repeated And Non-Repeated Character In A String?

*Step 1* : Define one HashMap called *charCountMap* with *Character* as key and *Integer* as value. This map will hold the characters and their count in the given string.

*Step 2* : Convert *inputString* to char array called *strArray*.

*Step 3* : Iterate through all chars of *strArray* and update their occurrences in *charCountMap*.

*Step 4* : Iterate through all chars of *strArray* and check their count in *charCountMap*. Any first occurring character with 1 as it’s count will be the first non-repeated character in *inputString*.

*Step 5* : Iterate through all chars of *strArray* and check their count in *charCountMap*. Any first occurring character with >1 as it’s count will be the first repeated character in *inputString*.

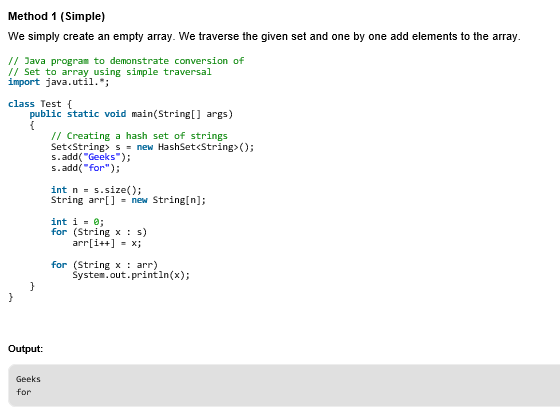


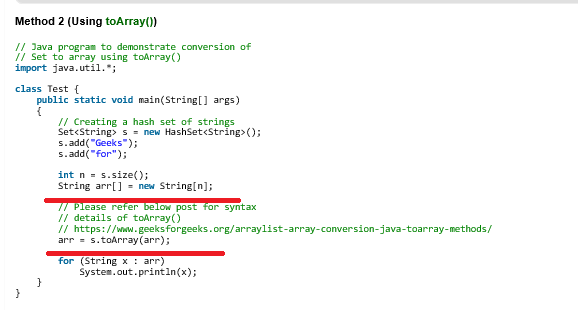
NOTE 🡺 IS THIS MEANS MAP MAINTAINS THE INSERTION ORDER

1. **Write a java program to append a given string to a text file?**

**MS 🡺 USE SOME FILEWRITER OBJECT AND WRITE INTO IT**

Tutorial from website 🡺https://www.geeksforgeeks.org/set-to-array-in-java/

1. Set to Array in Java  
   

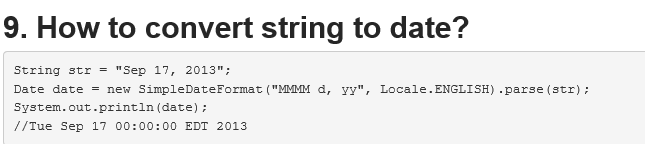
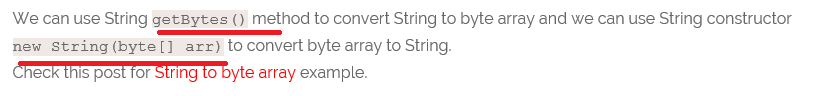
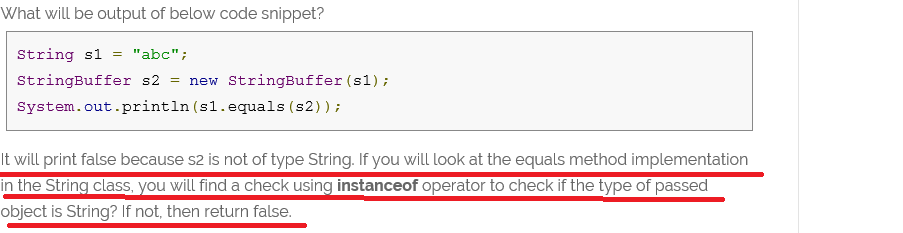


Questions from Journal Dev

1. Yes to version 7. From [JDK 7](http://openjdk.java.net/projects/jdk7/features/), we can use string as switch condition. Before version 6, we can not use string as switch condition.

## How to convert string to int?

int n = Integer.parseInt("10");

1. 
2. 
3. Write a method to check if input String is Palindrome? 🡺 we can use the api 🡺strBuilder.reverse();
4. How can we make String upper case or lower case? We can use String class toUpperCase and toLowerCase methods
5. What is String subSequence method? 🡺 Java 1.4 introduced CharSequence interface and String implements this interface, this is the only reason for the implementation of subSequence method in String class. Internally it invokes the String substring method.
6. How to convert String to char and vice versa? 🡺toCharArray
7. How to convert String to byte array and vice versa? 🡺
8. Why String is popular HashMap key in Java? Since String is immutable, its hashcode is cached at the time of creation and it doesn’t need to be calculated again.
9. 
10. 
11. 

<https://beginnersbook.com/2014/01/java-program-to-check-prime-number/>

1. Program to check whether input number is prime or not🡺
2. 