1) Write a java program to reverse a string?

```
class CountCharacterOccurence
{
   public static void main(String[] args)
   {
//Using StringBuffer class

        StringBuffer sbf = new StringBuffer("MyJava");
        System.out.println(sbf.reverse());          //Output : avaJyM

//Using iterative method
        String str = "MyJava";
        char[] strArray = str.toCharArray();
        for (int i = strArray.length - 1; i >= 0; i--)
        {
                System.out.print(strArray[i]);    //Output : avaJyM
        }
//Using recursive method.
        static String recursiveMethod(String str)
        {
                if ((null == str) || (str.length() <= 1))
                 {
                    return str;
                 }
                return recursiveMethod(str.substring(1)) + str.charAt(0);
        }

}
}
```

2) How do you convert string to integer and integer to string in java?

```
public class StringToInteger
{
   public static void main(String[] args)
   {
        String s = "2015";

        // Using Integer.parseInt() method
        int i1 = Integer.parseInt(s);
         System.out.println(i1);        //Output : 2015

        // Using Integer.valueOf() method
         int i 2= Integer.valueOf(s);
```

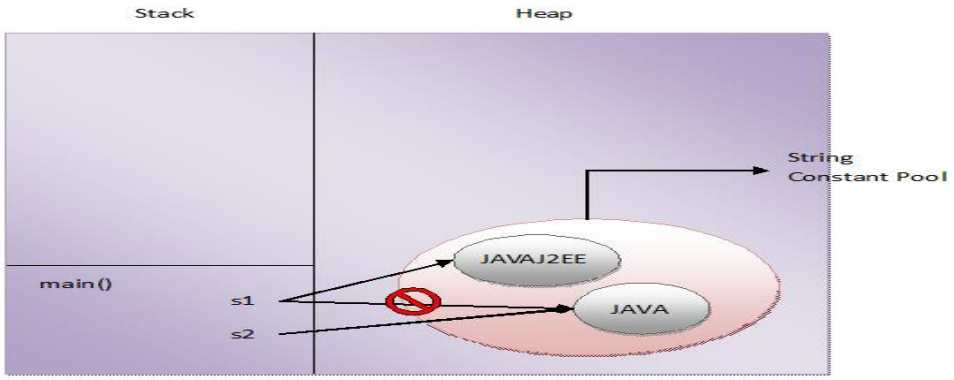
```
        System.out.println(i2);        //Output : 2015

        // Convert Integer To String
        // Using Integer.toString() Method
        int j=2015;
        String str1 = Integer.toString(j);
        System.out.println(str1);      //Output : 2015



        // Using Integer.valueOf() method
        String str2 = String.valueOf(i);
        System.out.println(str2);      //Output : 2015
        }
}
```

3) Write a java program to prove that strings are immutable in java?




```
public class StringExamples
{
   public static void main(String[] args)
   {
        String s1 = "JAVA";
        String s2 = "JAVA";
        System.out.println(s1 == s2);       //Output : true

        s1 = s1 + "J2EE";
        System.out.println(s1 == s2);       //Output : false
   }
}
```

That means now both s1 and s2 are pointing to two different objects in the pool. Before modifications they are pointing to same object. Once we tried to change the content of the object using 's1', a new object is created in the pool with "JAVAJ2EE" as it's content and it's reference is assigned to s1. If the strings are mutable, both s1 and

s2 should point to same object even after modification. That never happened here. That proves the string objects are immutable in java.

Are string objects created using new operator also immutable? The answer is Yes. String objects created using new operator are also immutable although they are stored in the heap memory.

```
public class StringExamples
{
   public static void main(String[] args)
   {
            String s1 = new String("JAVA");
            System.out.println(s1);        //Output : JAVA
           s1.concat("J2EE");
           System.out.println(s1);        //Output : JAVA
   }
}
```

4) Write a java program to count the number of words in a string?

```
class CountTheWords
{
   public static void main(String[] args)
   {
           System.out.println("Enter the string");
           Scanner sc = new Scanner(System.in);
           String s=sc.nextLine();
           String[] words = s.trim().split(" ");
           System.out.println("Number of words in the string = "+words.length);

           //---------------------------------- OR ---------------------------------------
           int count = 1;
            for (int i = 0; i < s.length()-1; i++)
            {
                    if((s.charAt(i) == ' ') && (s.charAt(i+1) != ' '))
                    {
                            count++;
                     }
           }
           System.out.println("Number of words in a string = "+count);

   }
}
```

5) Write a java program to find the duplicate words and their number of occurrences in a string?

```
public class duplicateWordsInString{
   static void duplicateWords(String inputString)
   {
      //Splitting inputString into words
           String[] words = inputString.split(" ");

      //Creating one HashMap with words as key and their count as value
       HashMap<String, Integer> wordCount = new HashMap<String, Integer>();

      //Checking each word
      for (String word : words)
      {
         //whether it is present in wordCount
              if(wordCount.containsKey(word.toLowerCase()))
            {
            //If it is present, incrementing it's count by 1
             wordCount.put(word.toLowerCase(), wordCount.get(word.toLowerCase())+1);
         }
         else
         {
            //If it is not present, put that word into wordCount with 1 as it's value
            wordCount.put(word.toLowerCase(), 1);
         }
      }

      //Extracting all keys of wordCount
           Set<String> wordsInString = wordCount.keySet();

      //Iterating through all words in wordCount

      for (String word : wordsInString)
      {
         //if word count is greater than 1
           if(wordCount.get(word) > 1)
          {
            //Printing that word and it's count
            System.out.println(word+" : "+wordCount.get(word));
         }
      }
   }
```

```
   public static void main(String[] args)
   {
         duplicateWords("Bread butter and bread");

         duplicateWords("Java is java again java");
         duplicateWords("Super Man Bat Man Spider Man");
   }
}
```

6) Write a java program to count the total number of occurrences of a given character in a string without using any loop?

```
class CountCharacterOccurence
{
   public static void main(String[] args)
   {
      String s = "Java is java again java again";
         char c = 'a';
          int count = s.length() - s.replace("a", "").length();
         System.out.println("Number of occurances of 'a' in "+s+" = "+count);
   }
}
```

7) Write a java program to count the number of occurrences of each character in a string?

```
class EachCharCountInString
{
         static void characterCount(String inputString)
         {
                  //Creating a HashMap containing char as a key and occurrences as  a value
                  HashMap<Character, Integer> charCountMap = new HashMap<Character, Integer>();

                  //Converting given string to char array
                   char[] strArray = inputString.toCharArray();

                  //checking each char of strArray
                  for (char c : strArray)
                   {
                           if(charCountMap.containsKey(c))
                           {
                                    //If char is present in charCountMap, incrementing it's count by 1
```

```
                                charCountMap.put(c, charCountMap.get(c)+1);
                        }
                        else
                        {
        //If char is not present in charCountMap,putting this char to charCountMap with 1 as it's value
                                charCountMap.put(c, 1);
                        }
                }

                        //Printing the charCountMap
                        System.out.println(charCountMap);
        }

   public static void main(String[] args)
   {
        characterCount("Java J2EE Java JSP J2EE");
         characterCount("All Is Well");
        characterCount("Done And Gone");
   }
}
```

8) Write a java program to find duplicate characters in a string?

```
import java.util.HashMap;
import java.util.Set;

class DuplicateCharactersInString
{
        static void duplicateCharCount(String inputString)
        {
                //Creating a HashMap containing char as key and it's occurrences as value
                HashMap<Character, Integer> charCountMap = new HashMap<Character, Integer>();

                //Converting given string to char array
                 char[] strArray = inputString.toCharArray();

      //checking each char of strArray
         for (char c : strArray)
         {
                        if(charCountMap.containsKey(c))
                        {
                                //If char is present in charCountMap, incrementing it's count by 1
                                charCountMap.put(c, charCountMap.get(c)+1);
```

```
                        }
                        else
                        {
        //If char is not present in charCountMap,putting this char to charCountMap with 1 as it's value
                                charCountMap.put(c, 1);
                        }
                }

    //Getting a Set containing all keys of charCountMap
        Set<Character> charsInString = charCountMap.keySet();
        System.out.println("Duplicate Characters In "+inputString);

    //Iterating through Set 'charsInString'
         for (Character ch : charsInString)
        {
                        if(charCountMap.get(ch) > 1)
                        {
                                //If any char has a count of more than 1, printing it's count
                                System.out.println(ch +" : "+ charCountMap.get(ch));
                        }
                }
        }
  public static void main(String[] args)
   {
           duplicateCharCount("JavaJ2EE");
        duplicateCharCount("Fresh Fish");
        duplicateCharCount("Better Butter");
   }
}
```

9) Write a java program to reverse each word of a given string?

```
public class ReverseEachWord
{
   static void reverseEachWordOfString(String inputString)
   {
        String[] words = inputString.split(" ");
        String reverseString = "";

        for (int i = 0; i < words.length; i++)
         {
                String word = words[i];
                String reverseWord = "";
                 for (int j = word.length()-1; j >= 0; j--)
                 {
```

```
                    reverseWord = reverseWord + word.charAt(j);
                }
                reverseString = reverseString + reverseWord + " ";
            }

        System.out.println(inputString);
        System.out.println(reverseString);
        System.out.println("-------------------------");
    }

    public static void main(String[] args)
    {
        reverseEachWordOfString("Java Concept Of The Day");
        reverseEachWordOfString("Java J2EE JSP Servlets Hibernate Struts");
        reverseEachWordOfString("I am string not reversed");
        reverseEachWordOfString("Reverse Me");
    }
}
```

10) Write a java program to find the percentage of uppercase letters, lowercase letters, digits and special characters in a given string?

Write a java program to find the percentage of uppercase letters, lowercase letters, digits and other special characters(including space) in the given string.

For example, In **"Tiger Runs @ The Speed Of 100 km/hour."**

Number of uppercase letters is **5**. So percentage is **13.16%**
Number of lowercase letters is **20**. So percentage is **52.63%**
Number of digits is **3**. So percentage is **7.89%**
Number of other characters is **10**. So percentage is **26.32%**

```
import java.text.DecimalFormat;

public class MainClass
{
    static void characterPercentage(String inputString)
    {
        //Getting total no of characters in the given string
        int totalChars = inputString.length();

        //Initializing upperCaseLetters, lowerCaseLetters, digits and others with 0
        int upperCaseLetters = 0;
        int lowerCaseLetters = 0;
        int digits = 0;
         int others = 0;

         //Iterating through each character of inputString
        for (int i = 0; i < inputString.length(); i++)
```

```
        {
                char ch = inputString.charAt(i);
                //If ch is in uppercase, then incrementing upperCaseLetters
                if(Character.isUpperCase(ch))
                {
                        upperCaseLetters++;
                }
                //If ch is in lowercase, then incrementing lowerCaseLetters
                else if(Character.isLowerCase(ch))
                {
                        lowerCaseLetters++;
                }
                //If ch is a digit, then incrementing digits
                else if (Character.isDigit(ch))
                {
                        digits++;
                }
                //If ch is a special character then incrementing others
                else
                {
                        others++;
                }
    }

    //Calculating percentage of uppercase letters, lowercase letters, digits and other characters
    double upperCaseLetterPercentage = (upperCaseLetters * 100.0) / totalChars ;
    double lowerCaseLetterPercentage = (lowerCaseLetters * 100.0) / totalChars;
    double digitsPercentage = (digits * 100.0) / totalChars;
    double otherCharPercentage = (others * 100.0) / totalChars;

    DecimalFormat formatter = new DecimalFormat("##.##");

    //Printing percentage of uppercase letters, lowercase letters, digits and other characters
     System.out.println("In '"+inputString+"' : ");
     System.out.println("Uppercase letters are "+formatter.format(upperCaseLetterPercentage)+"% ");
     System.out.println("Lowercase letters are "+formatter.format(lowerCaseLetterPercentage)+"%");
     System.out.println("Digits Are "+formatter.format(digitsPercentage)+"%");
     System.out.println("Other Characters Are "+formatter.format(otherCharPercentage)+"%");
     System.out.println("-----------------------------");
  }

  public static void main(String[] args)
  {
     characterPercentage("Tiger Runs @ The Speed Of 100 km/hour.");
     characterPercentage("My e-mail : eMail_Address321@anymail.com");
     characterPercentage("AUS : 123/3, 21.2 Overs");
  }
}
```

11) Write a java program to remove all white spaces from a string?

```
class RemoveWhiteSpaces
{
   public static void main(String[] args)
   {
      String str = "  Core Java jsp servlets           jdbc struts hibernate spring  ";

      //1. Using replaceAll() Method
            String strWithoutSpace = str.replaceAll("\\s", "");
           System.out.println(strWithoutSpace);

      //2. Without Using replaceAll() Method
            char[] strArray = str.toCharArray();
           StringBuffer sb = new StringBuffer();

      for (int i = 0; i < strArray.length; i++)
      {
         if( (strArray[i] != ' ') && (strArray[i] != '\t') )
         {
            sb.append(strArray[i]);
         }
      }
 System.out.println(sb);
   }
}
```

12) Write a java program to check whether one string is a rotation of another?

If s1 and s2 are two given strings, then write a java program to check whether s2 is a rotated version of s1?

Examples:

If **"JavaJ2eeStrutsHibernate"** is a string then below are some rotated versions of this string.
"StrutsHibernateJavaJ2ee", **"J2eeStrutsHibernateJava"**, **"HibernateJavaJ2eeStruts"**.

Solution:

Step 1 : Check whether s1 and s2 are of same length. If they are not of same length then s2 is not rotated version of s1.

Step 2 : s3 = s1 + s1;

If s1 = "JavaJ2eeStrutsHibernate" then s3 = "JavaJ2eeStrutsHibernateJavaJ2eeStrutsHibernate".

Step 3 : Check whether s3 contains s2 using contains() method of String class. If it contains then s2 is rotated version of s1.

```
public class MainClass
{
```

```
   public static void main(String[] args)
   {
          String s1 = "JavaJ2eeStrutsHibernate";
          String s2 = "StrutsHibernateJavaJ2ee";

          //Step 1
          if(s1.length() != s2.length())
          {
                  System.out.println("s2 is not rotated version of s1");
          }
          else
          {
          //Step 2
                  String s3 = s1 + s1;
          //Step 3
                 if(s3.contains(s2))
                 {
                         System.out.println("s2 is a rotated version of s1");
                  }
                  else
                 {
                         System.out.println("s2 is not rotated version of s1");
                  }
          }
   }
}
```

13) 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.

```
public class AnagramProgram
{
   static void isAnagram(String s1, String s2)
   {
      //Removing all white spaces from s1 and s2
          String copyOfs1 = s1.replaceAll("\\s", "");
          String copyOfs2 = s2.replaceAll("\\s", "");



     //Initially setting status as true
          boolean status = true;
          if(copyOfs1.length() != copyOfs2.length())
          {
                  //Setting status as false if copyOfs1 and copyOfs2 doesn't have same length
```

```
                status = false;
            }
             else
             {
        //Changing the case of characters of both copyOfs1 and copyOfs2 & converting them to char array

                char[] s1Array = copyOfs1.toLowerCase().toCharArray();
                char[] s2Array = copyOfs2.toLowerCase().toCharArray();

                 //Sorting both s1Array and s2Array
                 Arrays.sort(s1Array);
                 Arrays.sort(s2Array);

                 //Checking whether s1Array and s2Array are equal
                 status = Arrays.equals(s1Array, s2Array);
            }

        //Output
             if(status)
             {
                System.out.println(s1+" and "+s2+" are anagrams");
             }
             else
             {
                System.out.println(s1+" and "+s2+" are not anagrams");
             }
    }
 public static void main(String[] args)
    {
           isAnagram("Mother In Law", "Hitler Woman");
            isAnagram("keEp", "peeK");
            isAnagram("SiLeNt CAT", "LisTen AcT");
            isAnagram("Debit Card", "Bad Credit");
            isAnagram("School MASTER", "The ClassROOM");
            isAnagram("DORMITORY", "Dirty Room");
            isAnagram("ASTRONOMERS", "NO MORE STARS");
            isAnagram("Toss", "Shot");
            isAnagram("joy", "enjoy");
    }
}
```

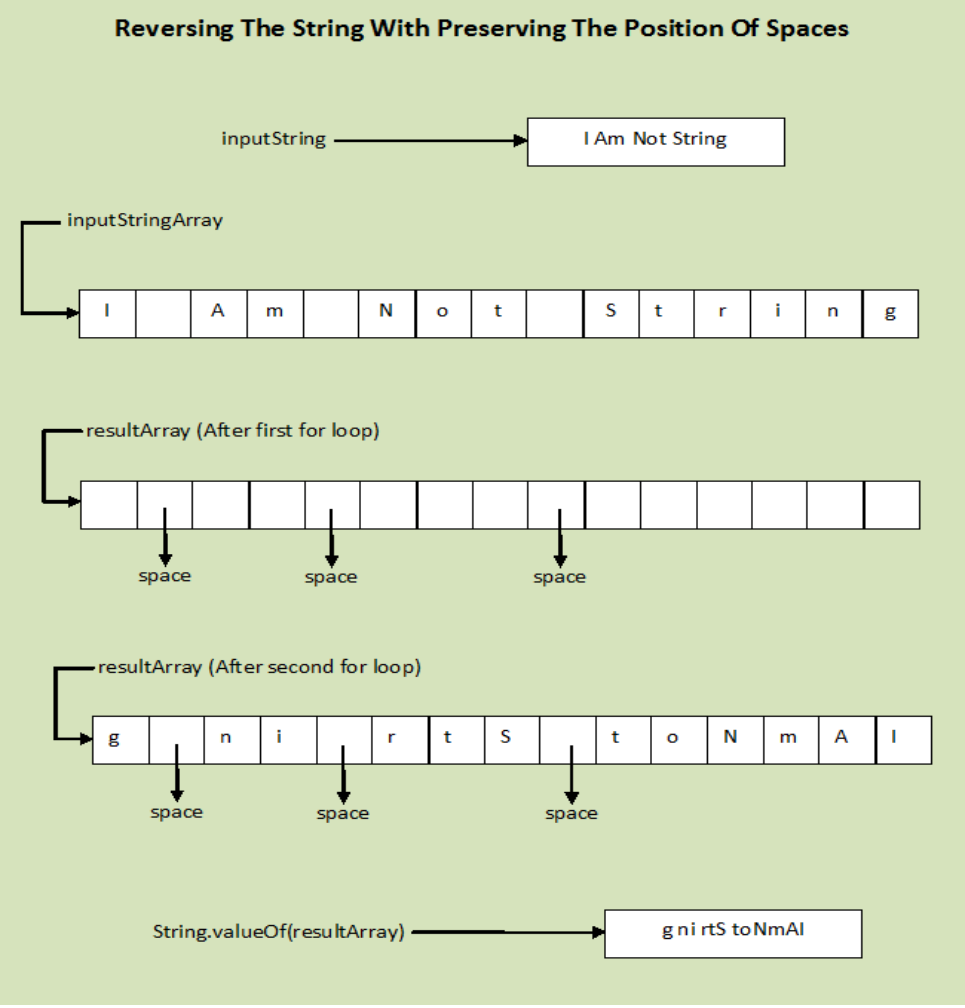
14) Write a java program to reverse a given string with preserving the position of spaces?

Write a java program to reverse a string with preserving the position of spaces.

For example, if **"I Am Not String"** is the given string then the reverse of this string with preserving the position of spaces is **"g ni rtS toNmAI"**.

Notice that the position of spaces in the original string are maintained in the reversed string also. That means, if the given string has a space at index 3, then there should be also a space in the reversed string at index 3.

- 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.



```
public class MainClass
{
   static void reverseString(String inputString)
   {
         //Converting inputString to char array 'inputStringArray'
         char[] inputStringArray = inputString.toCharArray();

         //Defining a new char array 'resultArray' with same size as inputStringArray
         char[] resultArray = new char[inputStringArray.length];

         //First for loop :
         /*For every space in the 'inputStringArray', we insert spaces in the 'resultArray' at the
         corresponding positions */

         for (int i = 0; i < inputStringArray.length; i++)
         {
                  if (inputStringArray[i] == ' ')
                 {
                           resultArray[i] = ' ';
                  }
      }

         //Initializing 'j' with length of resultArray
          int j = resultArray.length-1;

         //Second for loop :
         /* We copy every non-space character of inputStringArray from first to last at 'j' position of
         resultArray */

          for (int i = 0; i < inputStringArray.length; i++)
         {
                 if (inputStringArray[i] != ' ')
                  {
                           //If resultArray already has space at index j then decrementing 'j'
                          if(resultArray[j] == ' ')
                          {
                                   j--;
                          }
                  resultArray[j] = inputStringArray[i];
                  j--;
                  }
         }

      System.out.println(inputString+" ---> "+String.valueOf(resultArray));
   }

   public static void main(String[] args)
   {
```

```
        reverseString("I Am Not String");
        reverseString("JAVA JSP ANDROID");
         reverseString("1 22 333 4444 55555");
    }
}
```

15) Write a java program to find all permutations of a string?

```
public class PermutationsOfString
{
   static public void StringPermutation(String input)
   {
      StringPermutation("", input);
   }
   private static void StringPermutation(String permutation, String input)
   {
      if(input.length() == 0)
      {
                        System.out.println(permutation);
            }
      else
      {
         for (int i = 0; i < input.length(); i++)
         {
            StringPermutation(permutation+input.charAt(i), input.substring(0, i)+input.substring(i+1, input.length()));
         }
      }
   }

   public static void main(String[] args)
   {
      StringPermutation("JSP");
   }
}
```

How to compare strings?
How to search last occurrence of a substring inside a substring?
How to remove a particular character from a string?
How to replace a substring inside a string by another one?
How to reverse a String?
How to search a word inside a string?
How to split a string into a number of substrings?
How to convert a string totally into upper case?
How to match regions in a string?
How to compare performance of two strings?
How to optimize string creation?
How to format strings?
How to concatenate two strings?
How to get Unicode of strings?
How to buffer strings?