#String #interview #java #programs

------------------------------

1. **how to reverse a string in java without using any API.**

class reverseWithoutAPI   
{   
public static void main(String a[])   
{   
String str="Java 2 career";   
String arr[]=str.split(" ");   
System.out.print("Reversed String of "+str+":");   
for(int i=arr.length-1;i>=0;i--)   
{   
char temp[]=arr[i].toCharArray();   
for(int j=temp.length-1;j>=0;j--)   
{   
System.out.print(temp[j]);   
  
}   
System.out.print(" ");   
}   
}   
  
}

output- Reversed String of Java 2 career:reerac 2 avaJ

2**.> how to find duplicate charcater(no of occurance) in a string in java.**

1. **import** java.util.HashMap;
2. **import** java.util.Map;
3. **import** java.util.Set;
5. **public** **class** DuplicateCharFinder {
6. **public** **void** findIt(String str) {
7. Map<Character, Integer> baseMap = **new** HashMap<Character, Integer>();
8. **char**[] charArray = str.toCharArray();
9. **for** (Character ch : charArray) {
10. **if** (baseMap.containsKey(ch)) {
11. baseMap.put(ch, baseMap.get(ch) + 1);
12. } **else** {
13. baseMap.put(ch, 1);
14. }
15. }
16. Set<Character> keys = baseMap.keySet();
17. **for** (Character ch : keys) {
18. **if** (baseMap.get(ch) > 1) {
19. System.out.println(ch + "  is " + baseMap.get(ch) + " times");
20. }
21. }
22. }
24. **public** **static** **void** main(String a[]) {
25. DuplicateCharFinder dcf = **new** DuplicateCharFinder();
26. dcf.findIt("India is my country");
27. }
28. }

Output:

is 3 times

i is 2 times

n is 2 times

y is 2 times

1. **how to count occurance of each character in a string in java.**

|  |
| --- |
| 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");      }  } |

**Output :**

{E=4, 2=2, v=2, =4, P=1, S=1, a=4, J=5}  
{W=1, =2, e=1, s=1, A=1, l=4, I=1}  
{D=1, d=1, =2, G=1, e=2, A=1, n=3, o=2}

**Note :**

Above program is a case sensitive i.e it treats **‘A’** and **‘a’** as two different characters. If you want your program not to be case sensitive, convert the input string to either lowercase or uppercase using **toLowerCase()** or **toUpperCase()** methods.

1. **how do you remove all white spaces from a string in java.**
2. **public** **class** RemoveAllSpace {
3. **public** **static** **void** main(String[] args) {
4. String str = "India     Is My    Country";
5. //1st way
6. String noSpaceStr = str.replaceAll("\\s", ""); // using built in method
7. System.out.println(noSpaceStr);
8. //2nd way
9. **char**[] strArray = str.toCharArray();
10. StringBuffer stringBuffer = **new** StringBuffer();
11. **for** (**int** i = 0; i < strArray.length; i++) {
12. **if** ((strArray[i] != ' ') && (strArray[i] != '\t')) {
13. stringBuffer.append(strArray[i]);
14. }
15. }
16. String noSpaceStr2 = stringBuffer.toString();
17. System.out.println(noSpaceStr2);
18. }
19. }

Output:

IndiaIsMyCountry

IndiaIsMyCountry

4**.> how to check given String is palindrome or not**

**package** com.includehelp.stringsample;

**import** java.util.Scanner;

/\*\*

**\*** Easiest way to check Given String is Palindrome String or not

**\*** **@author** **includehelp**

 \*/

**public** **class** PalindromString {

**static** boolean isPalindromString(**String** inputStr){

StringBuilder sb = **new** StringBuilder(inputStr);

**String** reverseStr = sb.reverse().toString();

**return** (inputStr.equalsIgnoreCase(reverseStr));

}

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

Scanner sc = **new** Scanner(**System**.in);

**System**.out.println("Enter String : ");

**String** inString = sc.next();

**if**(isPalindromString(inString)){

**System**.out.println(inString +" is a Palindrom String");

}

**else**{

**System**.out.println(inString +" is not a Palindrom String");

}

}

}

Output

First run:

Enter String : india

india is not a Palindrom String

Second run:

Enter String : abcba

abcba is a Palindrom String

5**.> how to check if two String are anargms.**

|  |
| --- |
| // JAVA program to check whether two strings  // are anagrams of each other  import java.io.\*;  import java.util.Arrays;  import java.util.Collections;    class GFG {        /\* function to check whether two strings are      anagram of each other \*/      static boolean areAnagram(char[] str1, char[] str2)      {          // Get lenghts of both strings          int n1 = str1.length;          int n2 = str2.length;            // If length of both strings is not same,          // then they cannot be anagram          if (n1 != n2)              return false;            // Sort both strings          Arrays.sort(str1);          Arrays.sort(str2);            // Compare sorted strings          for (int i = 0; i < n1; i++)              if (str1[i] != str2[i])                  return false;            return true;      }        /\* Driver program to test to print printDups\*/      public static void main(String args[])      {          char str1[] = { 't', 'e', 's', 't' };          char str2[] = { 't', 't', 'e', 'w' };          if (areAnagram(str1, str2))              System.out.println("The two strings are"                                 + " anagram of each other");          else              System.out.println("The two strings are not"                                 + " anagram of each other");      }  }    // This code is contributed by Nikita Tiwari. |

**Output:**

The two strings are not anagram of each other

6**.> how to find duplicate character in a String.**

public class DuplStr {

public static void main(String argu[]) {

String str = "w3schools";

int cnt = 0;

char[] inp = str.toCharArray();

System.out.println("Duplicate Characters are:");

for (int i = 0; i < str.length(); i++) {

for (int j = i + 1; j < str.length(); j++) {

if (inp[i] == inp[j]) {

System.out.println(inp[j]);

cnt++;

break;

}

}

}

}

}

Program Output:

Duplicate Characters are: s o