

**1.What is Mutable String in Java Explain with an example.**

**=>** ***Mutable String***

Once if we create a String, on that String if we try to perform any operation and if those changes get reflected in the same object then such strings are called “Mutable String”.

Example: StringBuffer, StringBuilder

We can use the StringBuffer and StringBuilder classes to generate mutable strings. Which class we should use entirely depends on the scenario. Both classes generate a mutable object of string.   
If the string needs to be thread-safe and you wish to operate in a multithreading environment, you should use the StringBuffer class. On the other hand, StringBuilder is not necessary if you don't want a multithreading environment.

However, if speed is your top priority, StringBuilder outperforms StringBuffer in terms of speed.

***Example***   
StringBuffer sb = new StringBuffer();

System.out.println(sb.capacity());//16

sb.append("abcdefghijklmnop");

System.out.println(sb.capacity());//16

sb.append('q');

System.out.println(sb.capacity());//34

**2.WAP to reverse a String**

**Input: “PWSKILLS”**

**Output: “SLLIKSPW”**

**=>** public class ReversingString {  
 public static void main(String[] args) {  
 System.*out*.print("Enter the String : ");  
 Scanner sc=new Scanner(System.*in*);  
 String str1 =sc.nextLine();  
 String str2="";  
 for (int i = 0; i < str1.length(); i++) {  
 str2=str2+(str1.charAt(str1.length()-(i+1)));  
 }  
 System.*out*.println(str2);  
 }  
}

**3.WAP to reverse a sentence while preserving the position.**

**Input: Think Twice**

**Output: “knihT eciwT”**

**=>** public class ReversingString {  
 public static void main(String[] args) {  
 Scanner sc=new Scanner(System.*in*);  
 System.*out*.print("Enter the String : ");  
 String str1=sc.nextLine();  
 String str2="";  
 String str[]=str1.split(" ");  
 for (String s:str){  
 for (int i = 0; i < s.length(); i++) {  
 str2=str2+s.charAt(s.length()-(i+1));  
 }  
 str2=str2+" ";  
 }  
 System.*out*.println(str2);  
 }  
}

**4.WAP to sort a String Alphabetically .**

**=>** public class sortString {  
 public static void main(String[] args) {  
 Scanner sc=new Scanner(System.*in*);  
 System.*out*.print("Enter the String : ");  
 String str1=sc.nextLine();  
 str1=str1.toLowerCase();  
 char ch[]=str1.toCharArray();  
 Arrays.*sort*(ch);  
 for (int i = 0; i < ch.length; i++) {  
 System.*out*.println(ch[i]);  
  
 }  
 }  
}