

# Difference between String and StringBuffer

There are many differences between String and StringBuffer. A list of differences between String and StringBuffer are given below:

No.	String	StringBuffer
1)	String class is immutable.	StringBuffer class is mutable.
2)	String is slow and consumes more memory when you concat too many strings because every time it creates new instance.	StringBuffer is fast and consumes less memory when you concat strings.
3)	String class overrides the equals() method of Object class. So you can compare the contents of two strings by equals() method.	StringBuffer class doesn't override the equals() method of Object class.

## Performance Test of String and StringBuffer

```
public class ConcatTest{
```

```
public static String concatWithString()  {  
    String t = "Java";  
    for (int i=0; i<10000; i++){  
        t = t + "Tpoint";  
    }  
    return t;  
}  
  
public static String concatWithStringBuffer(){  
    StringBuffer sb = new StringBuffer("Java");  
    for (int i=0; i<10000; i++){  
        sb.append("Tpoint");  
    }  
    return sb.toString();  
}  
  
public static void main(String[] args){  
    long startTime = System.currentTimeMillis();  
    concatWithString();  
    System.out.println("Time taken by Concating with String: "+  
(System.currentTimeMillis()-startTime)+"ms");  
    startTime = System.currentTimeMillis();  
    concatWithStringBuffer();  
    System.out.println("Time taken by Concating with  StringBuffer: "+  
(System.currentTimeMillis()-startTime)+"ms");  
}  
}
```

Time taken by Concating with String: 578ms

Time taken by Concating with StringBuffer: 0ms

## String and StringBuffer hashCode Test

As you can see in the program given below, String returns new hashcode value when you concat string but StringBuffer returns same.

```
public class InstanceTest{  
    public static void main(String args[]){  
        System.out.println("Hashcode test of String:");  
    }  
}
```

```
String str="java";
System.out.println(str.hashCode());
str=str+"tpoint";
System.out.println(str.hashCode());

System.out.println("Hashcode test of StringBuffer:");
StringBuffer sb=new StringBuffer("java");
System.out.println(sb.hashCode());
sb.append("tpoint");
System.out.println(sb.hashCode());
}
}
```

Hashcode test of String:

3254818

229541438

Hashcode test of StringBuffer:

118352462

118352462

[< prev](#)[next >](#)

## Share this page



## Latest 4 Tutorials



CouchDB



Docker



Rails



RichFaces