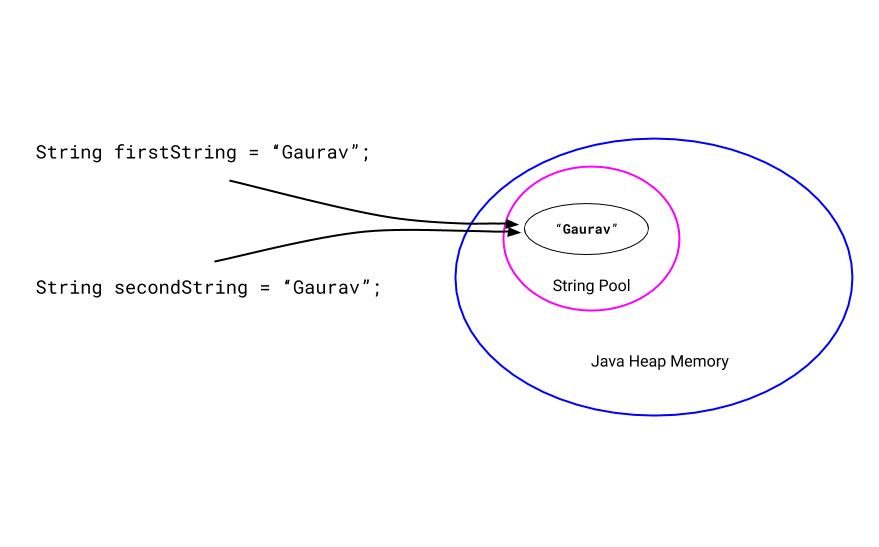
Top 10 Java String Interview Questions And Answers

### **1. What is String in Java? Is it a datatype?**

**Ans =** The string is a final class in Java defined in java.lang package. You can assign a sequence of characters to a string variable.  
 For example String name = "Gaurav";

### **2. What is the String pool in Java?**

**Ans=** The String pool is a special type of memory maintained by the JVM ,  
String pool is used to store unique string objects.

Ex :- When you assign the same string literal to different string variables, JVM saves only one copy of the String object in the String pool, and String variables will start referring to that string object.  
  
show with diagram:  
  
  
  
 Two string variables pointing to the single string object from the string pool

### **3. Why String is immutable? Ans=** Java String pool is possible because the String is immutable. If you assign the same string literal to many string variables, JVM will save only one copy of the string object in the Java string pool, and these variables will start referring to that string object.

**4. Suppose we have three string variables:-**

|  |  |
| --- | --- |
| 1  2  3 | String firstString = "Gaurav";  String secondString = "Gaurav";  String thirdString =  new String("Gaurav"); |

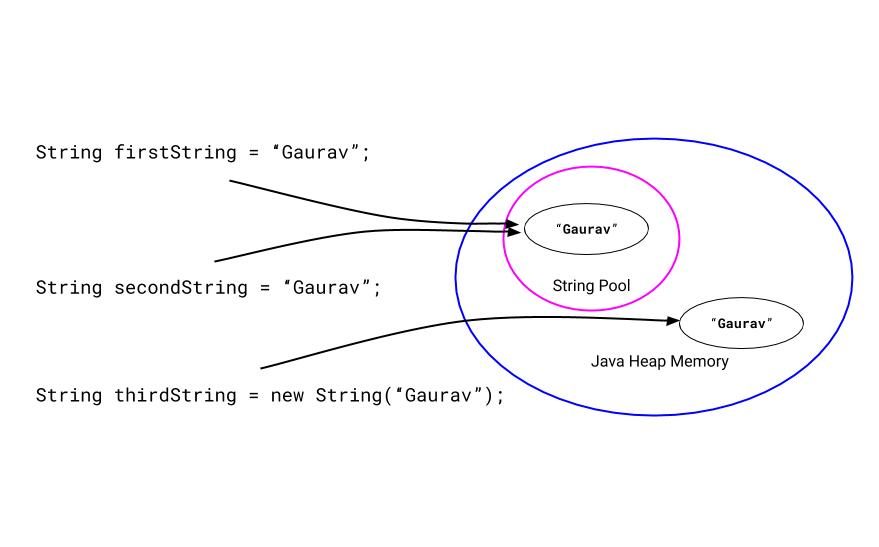
**🡪How many objects will be created from the following code?**

**Ans= By seeing the above code, only two string objects will be created.**   
  
The first two variables will refer to the same string object with the value "Gaurav". JVM uses the string pool concept to store only one copy of duplicate string objects to string constant pool.

But when we use a new keyword to create a new string, a new string object will be created and stored in the Java heap memory.

So for the third variable thirdString, a new string object will be created and stored in a Java heap space.

So there will be a total of two objects, one from the Java string pool and one from the Java heap memory.

Below, I have shown these two objects in the following diagram.  
 

### **5. What is the difference between the String and StringBuffer?**

**Ans =** The String is a final class in Java. The String is immutable. That means we can not change the value of the String object afterword.

Since the string is widely used in applications, we have to perform several operations on the String object. Which generates a new String object each time, and all previous objects will be garbage object putting the pressure on the Garbage collector.

Hence, the Java team introduced the StringBuffer class. It is a mutable String object, which means you can change its value.

The string is immutable, but the StringBuffer is mutable.

### **6. Can we compare String using the == operator? What is the risk? Ans=** Yes, of course, we can compare String using the == operator. But when we are comparing string using the == operator, we are comparing their object reference

**Program:-**

public class StringCompareUsingEqualsOperator {

public static void main(String[] args) {

String firstString = "Gaurav";

String secondString = "Gaurav";

String thirdString =  new String("Gaurav");

System.out.print("Case 1 : ");

System.out.println(firstString == secondString); // true

System.out.print("Case 2 : ");

System.out.println(firstString == thirdString); // false

// Comparing strings using equals() method

System.out.print("Case 3 : ");

System.out.println(firstString.equals(thirdString)); // true

}

}

The output of the above program will be:

|  |  |
| --- | --- |
| 1  2  3 | Case 1 : true  Case 2 : false  Case 3 : true |

**7. What are the ways to compare string?  
  
Ans=**We can compare strings using the equals() method, == operator and compareTo() method.  
When we compare strings using the equals() method, we are comparing the content of the strings, whether these strings have the same content or not.

When we compare strings using the == operator, we are comparing the reference of the string, whether these variables are pointing to the same string object or not.

Also, we can compare string lexicographically (comparing strings by alphabetical order). We can use the compareTo() method to compare the strings lexicographically.

### 

### **8. What is the use of the substring() method? Ans=** The substring() method in Java returns a ‘substring’ of the specified string.

There are two variants of the Substring method.

1. substring(int beginIndex)

2. substring(int beginIndex, int endIndex)

For 1st:  
String name = "Gaurav Kukade";

String result = name.substring(4);

System.out.println(result);

**Output: it will print “av kukade”.**

For 2nd:  
String name = "Gaurav Kukade";

String result = name.substring(4,9);

System.out.println(result);

### **Output: it will print “av ku”.**

**9. How to check if the String is empty?  
Ans=** Java String class has a special method to check if the string is empty or not.  
  
isEmpty() method internally checks if the length of the string is zero. If it is zero, that means the string is empty, and the isEmpty() method will return true.

If the length of the string is not zero, then the isEmpty() method will return false.