What is String ?

Ans... In Java, a string is an object that represents a number of character values. Each letter in the string is a separate character value that makes up the Java string object. Characters in Java are represented by the char class. Users can write an array of char values that will mean the same thing as a string.

Types of String?

Ans.....

Types of strings in Java

Primitive strings. These are string literals or string calls from a nonconstructor context. A constructor is a special method used to initialize objects. ...

Object strings. These are strings created using the new operator. Object strings create two objects, whereas primitives create just one.

Is string a data type in Java?

Definitely, String is not a primitive data type. It is a derived data type. Derived data types are also called reference types because they refer to an object. They call methods to perform operations.

How many types of strings are there in Java?

Types of String Classes

There are two types of character sequence classes in Java. The immutable class which is the String class, and the mutable class which is StringBuilder and StringBuffer. StringBuilder and StringBuffer have some differences. StringBuffer is thread-safe and synchronized.

Which datatype is string?

A string is generally considered a data type and is often implemented as an array data structure of bytes (or words) that stores a sequence of elements, typically characters, using some character encoding.

How many types of string are there?

There are two kinds of strings: variable-length and fixed-length strings

What are the classes of string?

Java String -

String, StringBuffer and StringBuilder classes implement it. It means, we can create strings in Java by using these three classes. The Java String is immutable which means it cannot be changed. Whenever we change any string, a new instance is created.

Why is it called a string?

Strings are called "strings" because they are made up of a sequence, or string, of characters.

Is string a variable type?

The two common types of variables that you are likely to see are numeric and string.

Why string is immutable in Java?

Strings in Java are specified as immutable, as seen above because strings with the same content share storage in a single pool to minimize creating a copy of the same value. That is to say, once a String is generated, its content cannot be changed and hence changing content will lead to the creation of a new String.

What are 3 different string instruments?

The most common string instruments in the string family are guitar, electric bass, violin, viola, cello, double bass, banjo, mandolin, ukulele, and harp

What are the 4 string functions in Java?

A number of methods provided in Java to perform operations in Strings are called String functions. The methods are compare(), concat(), equals(), split(), length(), replace(), compareTo() and so on.

What is Stringbuffer in Java?

A thread-safe, mutable sequence of characters. A string buffer is like a String , but can be modified. At any point in time it contains some particular sequence of characters, but the length and content of the sequence can be changed through certain method calls.

What is new string in Java?

For Example: String s="Welcome"; By new keyword : Java String is created by using a keyword "new". For example: String s=new String("Welcome"); It creates two objects (in String pool and in heap) and one reference variable where the variable 's' will refer to the object in the heap.

Is string a final class in Java?

The String class in the java. lang package is a final class for just this reason. The String class is so vital to the operation of the compiler and the interpreter that the Java system must guarantee that whenever a method or object uses a String they get exactly a java. lang

Why string is special in Java?

String is Special -

Java String is, however, special. Unlike an ordinary class: String is associated with string literal in the form of double-quoted texts such as " hello, world ". You can assign a string literal directly into a String variable, instead of calling the constructor to create a String instance.

Explain String pool in Java.

String Pool, also known as SCP (String Constant Pool), is a special storage space in Java heap memory that is used to store unique string objects. Whenever a string object is created, it first checks whether the String object with the same string value is already present in the String pool or not, and if it is available, then the reference to the string object from the string pool is returned. Otherwise, the new string object is added to the string pool, and the respective reference will be returned.

5. Is String immutable or final in Java? If so, then what are the benefits of Strings being Immutable?

Yes, Strings are immutable in Java. Immutable objects mean they can't be changed or altered once they've been created. However, we can only modify the reference to the string object. The String is immutable in Java because of many reasons like security, caching, synchronization and concurrency, and class loading

9. In Java, how can two strings be compared?

In Java, there are several ways for comparing two strings. The following are a few of them:

String Equals Method: In this method, the strings are compared based on the values within them. If the values of the two strings are the same, it returns true; otherwise, it returns false. This method is case-sensitive.

Syntax:

str1.equals(str2);

For example:

Input 1= Scaler

Input 2= InterviewBit

Output= false

Input 1= Scaler

Input 2= Scaler

Output= true

Input 1= Scaler

Input 2= scaler

Output= false

String Equals Ignore Case: By using this method, the two strings are compared without taking into account the case (upper or lower). It returns true if the two values are the same and not null.

Syntax:

str1.equalsIgnoreCase(str2);

For Example:

Input 1= Scaler

Input 2= InterviewBit

Output= false

Input 1= Scaler

Input 2= Scaler

Output= true

Input 1= Scaler

Input 2= scaler

Output= true

String Equals Ignore Case: By using this method, the two strings are compared without taking into account the case (upper or lower). It returns true if the two values are the same and not null.

Syntax:

str1.equalsIgnoreCase(str2);

For Example:

Input 1= Scaler

Input 2= InterviewBit

Output= false

Input 1= Scaler

Input 2= Scaler

Output= true

Input 1= Scaler

Input 2= scaler

Output= true

String Compare To Method: This method compares input strings with each other. Upon comparison, the following value is returned:

If (str1>str2), a positive value is returned.

If (str1==str2), 0 is returned.

If (str1<str2), a negative value is returned.

Syntax:

str1.compareTo(str2)

Example:

Input 1= InterviewBit

Input 2= Scaler

Output= -10

Input 1= Scaler

Input 2= Scaler

Output= 0

Input 1= Scaler

Input 2= InterviewBit

Output= 10

10. What is the difference between str1 == str2 and str1.equals(str2)?

Java offers both the equals() method and the "==" operator for comparing objects. However, here are some differences between the two:

Essentially, equals() is a method, while == is an operator.

The == operator can be used for comparing references (addresses) and the .equals() method can be used to compare content. To put it simply, == checks if the objects point to the same memory location, whereas .equals() compares the values of the objects.

Example:

```
public class StringComparison
{
  public static void main(String[] args)
  {
     String str1=new String("Scaler");
     String str2=new String("Scaler");
     System.out.println(str1 == str2);
     System.out.println(str1.equals(str2));
  }
}
```

Output:

```
false
true
```

12. What is the use of the substring() method in Java?

The substring method is used to return substring from a specified string. This method takes two parameters i.e., beginIndex (the starting index) and endIndex (the ending index). In the case of substring(), method startIndex is inclusive and endIndex is exclusive.

Syntax:

substring(int beginIndex, int endIndex)

Or

substring(int beginIndex)

Here,

beginIndex: Index that marks the starting of subsequence and it is inclusive.

endIndex: Index that marks the ending of subsequence and it is exclusive.

Example:

```
import java.lang.Math;
public class InterviewBit
{
  // driver code
  public static void main(String args[])
  {
      String str = "Scaler by InterviewBit";

     //prints substring from 7th index
      System.out.print("Returns: ");
      System.out.println(str.substring(7));
      // prints substring from 0-6, exclusive 6th index
      System.out.print("Returns: ");
```

```
        System.out.println(str.substring(0, 6));
       // prints the substring from 10-22, exclusive 22th index
        System.out.print("Returns: ");
        System.out.println(str.substring(10, 22));
  }
  }
```

Output:

```
Returns: by InterviewBit
Returns: Scaler
Returns: InterviewBi
```

13. Can we use a string in the switch case in java?

Yes, Java allows you to use strings in switch case conditions. Below is a Java program that shows the use of string in switch case.

Example:

```
public class StringinSwitchCase
{
  public static void main(String[] args)
  {
     String fruit = "Apple";
     switch(fruit)
     {
        case "Mango":
           System.out.println("Sweet");
           break;
        case "Apple":
```

```
            System.out.println("Delicious");
            break;
        case "Orange":
            System.out.println("Luscious");
            break;
        default:
            System.out.println("Not a fruit");
    }
  }
}
```

Output:

Delicious

1. What are the different string methods in Java?

There are various string operations in Java that allow us to work with strings. These methods or operations can be used for string handling in Java as well as string manipulation in Java. Some of such methods are as follows:

split(): Split/divide the string at the specified regex.

compareTo(): Compares two strings on the basis of the Unicode value of each string character.

compareToIgnoreCase(): Similar to compareTo, but it also ignores case differences.

length(): Returns the length of the specified string.

substring(): Returns the substring from the specified string.

equalsIgnoreCase(): Compares two strings ignoring case differences.

contains(): Checks if a string contains a substring.

trim(): Returns the substring after removing any leading and trailing whitespace from the specified string.

charAt(): Returns the character at specified index.

toLowerCase(): Converts string characters to lower case.

toUpperCase(): Converts string characters to upper case.

concat(): Concatenates two strings.

2. Is String thread-safe in Java?

Strings are immutable objects, which means they can't be changed or altered once they've been created. As a result, whenever we manipulate a String object, it creates a new String rather than modifying the original string object. In Java, every immutable object is thread-safe, which means String is also thread-safe. As a result, multiple threads can access a string. For instance, if a thread modifies the value of a string, instead of modifying the existing one, a new String is created, and therefore, the original string object that was shared among the threads remains unchanged

4. What is the best way to split a string in Java?

Split() is a Java method for breaking a string based on a Java string delimiter (specified regex). For example, a space or a comma(,) will usually be used as the Java string split attribute to break or split the string.

Syntax:

string.split(String regex, int limit)

Here,

regex: String is divided at this specified regex.

limit (optional parameter): Controls or limits the number of resulting substrings. Split() returns all potential substrings if the limit parameter is not specified or is 0.

Example:

```
public class SplitString
{
 public static void main(String[] args)
 {
   String str = "Scaler by InterviewBit";
```

```
  // split string from space
  String[] result = str.split(" ");
  for (int i=0; i < result.length; i++)
  {
    System.out.println(result[i]);
  }
 }
}
```

Output:

Scaler

by

InterviewBit