**String Assignment**

Asg Ques2

1.What is a String in Java

In Java, a **String** is a sequence of characters. It is a reference data type, which means it stores a reference (memory address) to the actual string data, rather than storing the data directly.

Strings in Java are immutable, which means once a string object is created, its value cannot be changed. If you need to modify a string, such as concatenating two strings together or converting it to uppercase, Java creates a new string object with the modified value.

**Example : String str = "Hello, World!";**

2.Types Of String in Java are?

1. **String**

String is an immutable class which means a constant and cannot be changed once created and if wish to change , we need to create an new object and even the functionality it provides like toupper, tolower, etc all these return a new object , its not modify the original object. It is automatically thread safe.

**Syntax**

**String str= "saurabh";  
 or  
String str= new String("saurabh")**

**2. StringBuffer**

[StringBuffer](https://www.geeksforgeeks.org/stringbuffer-class-in-java/) is a peer class of String that provides much of the functionality of strings. The string represents fixed-length, immutable character sequences while StringBuffer represents growable and writable character sequences means it is mutable in nature and it is thread safe class , we can use it when we have multi threaded environment and shared object of string buffer i.e, used by mutiple thread. As it is thread safe so there is extra overhead, so it is mainly used for multithreaded program.

**Syntax:**

**StringBuffer demoString = new StringBuffer("Saurabh");**

**3. StringBuilder**

[StringBuilder](https://www.geeksforgeeks.org/stringbuilder-class-in-java-with-examples/) in Java represents a mutable sequence of characters. Since the String Class in Java creates an immutable sequence of characters, the StringBuilder class provides an alternative to String Class, as it creates a mutable sequence of characters and it is not thread safe and its used only within the thread , so there is no extra overhead , so it is mainly used for single threaded program.

**Syntax:**

**StringBuilder demoString = new StringBuilder();  
demoString.append("Shiyeakr");**

3.In how many ways can you create string objects in java?

1. **String Literal**: A string literal is a sequence of characters enclosed in double quotes.

**For example:**

**String str = "Hello";**

1. **String Object**: A string object is an instance of the String class. You can create a string object using the new keyword.

**For example:**

**String str = new String("Hello");**

4.What is a string constant pool?

The string constant pool, also known as the string pool, is a special area in the Java heap memory that stores unique string literals. When you create a string using a string literal (e.g., "Hello"), Java checks the string pool to see if an identical string already exists. If it does, Java returns a reference to the existing string object from the pool instead of creating a new object. This helps conserve memory by reusing existing string objects.

**Or**

 A string constant pool is a separate place in the heap memory where the values of all the strings which are defined in the program are stored. When we declare a string, an object of type String is created in the stack, while an instance with the value of the string is created in the heap.

5.What do you mean by mutable and immutable objects?

he mutable objects are objects whose value can be changed after initialization. We can change the object's values, such as field and states, after the object is created. For example, [Java.util.Date](https://www.javatpoint.com/java-util-date)**,**[StringBuilder](https://www.javatpoint.com/StringBuilder-class)**,**[StringBuffer](https://www.javatpoint.com/StringBuffer-class), etc.

When we made a change in existing mutable objects, no new object will be created; instead, it will alter the value of the existing object. These object's classes provide methods to make changes in it.

The Getters and Setters ( get() and set() methods ) are available in mutable objects. The Mutable object may or may not be thread-safe.

Top of Form

The immutable objects are objects whose value can not be changed after initialization. We can not change anything once the object is created. For example, **primitive objects** such as [int](https://www.javatpoint.com/int-keyword-in-java), [long](https://www.javatpoint.com/long-keyword-in-java), [float](https://www.javatpoint.com/float-keyword-in-java), [double](https://www.javatpoint.com/double-keyword-in-java), **all**[legacy classes](https://www.javatpoint.com/legacy-class-in-java)**,**[Wrapper class](https://www.javatpoint.com/wrapper-class-in-java)**,**[String class](https://www.javatpoint.com/methods-of-string-class), etc.

In a nutshell, immutable means unmodified or unchangeable. Once the immutable objects are created, its object values and state can not be changed.

Only Getters ( get() method) are available not Setters ( set() method) for immutable objects.

6.Where exactly is the string constant pool located in the memory?

The string constant pool, or simply the string pool, is located in the Java heap memory. It is a special area within the heap memory where Java stores unique string literals.

The heap memory is a region of the computer's memory that is used for dynamic memory allocation. It is where objects (including strings) are allocated when they are created using the **new** keyword. The string pool is a part of this heap memory and is used specifically for storing string literals and interned strings.

Top of Form