# ASSIGNMENT

## Q1 ans-

The String class in Java is immutable, meaning that once a String object is created, its value cannot be changed.

## Q2 ans-

In Java, there are a few different types of strings or variations of the `String` class that you can work with. Let's explore some of these types:

1. **String**: The standard `String` class represents a sequence of characters. It is immutable, meaning its value cannot be changed once created.
2. **StringBuilder**: The `StringBuilder` class is a mutable alternative to `String`. It provides methods for efficient string manipulation when you need to perform a lot of modifications on a string. Unlike `String`, `StringBuilder` can be modified in-place without creating new objects, which can be more efficient when building or modifying strings.
3. **StringBuffer**: The `StringBuffer` class is similar to `StringBuilder` and also provides a mutable sequence of characters. It is thread-safe, unlike `StringBuilder`, which means it can be safely used in multi-threaded environments. However, it is generally recommended to use `StringBuilder` unless thread safety is explicitly required, as `StringBuffer` has some performance overhead due to synchronization.

## Q3 ans-

In Java, there are several ways to create a `String` object. Here are some of the common ways to create a `String` object:

1. **Using the new Keyword**: You can also create a `String` object using the `new` keyword along with the `String` class constructor:
2. **Using `StringBuilder` or `StringBuffer`**: If you need to perform dynamic string manipulation, you can use `StringBuilder` or `StringBuffer` and convert the resulting object to a `String` using the `toString()` method:

## Q4 ans-

In Java, the "String constant pool" is a special memory area in the Java Virtual Machine (JVM) where string literals are stored. It is an optimization technique used to reduce memory usage and improve performance when working with strings.

When you create a string using a string literal, such as `"Hello, World!"`, Java checks if an identical string already exists in the string constant pool. If it does, the existing string reference is returned instead of creating a new string object. This means that multiple references to the same string literal will point to the same memory location, saving memory.

## Q5 ans-

1. **Mutable Objects**: Mutable objects are objects whose state can be modified after they are created. This means that you can change the values of their fields or properties. The state of a mutable object can be modified by invoking methods or directly accessing its internal data. Examples of mutable objects in Java include `StringBuilder`
2. **Immutable Objects**: Immutable objects, on the other hand, are objects whose state cannot be changed after they are created. Once an immutable object is created, its internal state remains constant throughout its lifetime. In Java, the `String` class is an example of an immutable class.

## Q6 ans-

In Heap Memory.