

1.	What is a String in Java?
Ans.	String is a sequence of characters. In java programming, strings are objects.
2.	Types of Strings in Java are?
Ans.	<p>Primitive strings. These are string literals or string calls from a non-constructor context. A constructor is a special method used to initialize objects. Primitives are not objects and have no methods or properties. They are represented at the lowest level of language implementation.</p> <p>Object strings. These are strings created using the new operator. Object strings create two objects, whereas primitives create just one. Object strings create the string literal and the variable to refer to it.</p>
3.	In how many ways can you create string objects in Java?
Ans.	<p>String Literal: This is the most common and convenient way to create a string object. You can use double quotes to enclose the characters and create a string literal, like this: <code>String str = "Hello, World!";</code></p> <p>Using the new Keyword: You can create a string object using the new keyword and the String class constructor. For example: <code>String str = new String("Hello");</code></p> <p>Using StringBuilder or StringBuffer: If you need to build a string dynamically or perform frequent string manipulations, you can use <code>StringBuilder</code> or <code>StringBuffer</code> classes. These classes provide methods for efficient string concatenation and modification.</p>
4.	What is a string constant pool?
Ans.	<p>The string constant pool (also known as the string pool) is a special area in memory where string literals are stored. It is a part of the Java runtime environment and is used to optimize memory allocation and improve performance.</p> <p>When you create a string using a string literal, the Java compiler checks if the same string already exists in the string constant pool. If it does, the existing string reference is returned instead of creating a new object. This means that multiple string variables can refer to the same underlying string object in the pool.</p>

	<p>Memory Efficiency: Since string literals are reused in the pool, it helps conserve memory. Instead of creating multiple instances of the same string, only one instance is stored in memory.</p> <p>String Equality: String literals from the pool are compared for equality by reference. That means you can use the == operator to compare string literals and check if they refer to the same object in the pool.</p> <p>String Immutability: The immutability of strings is reinforced by the string constant pool. Since strings cannot be modified, multiple references to the same string ensure that its value remains unchanged.</p>
<p>5.</p> <p>Ans.</p>	<p>What do you mean by mutable and immutable objects?</p> <p>In programming, the terms "mutable" and "immutable" are used to describe the ability or inability to change an object after it has been created. Here's an explanation of what mutable and immutable objects mean:</p> <p>Mutable Objects: Mutable objects are objects whose state (i.e., their data) can be modified after they are created. This means you can change their properties, values, or internal state without creating a new object. Examples of mutable objects include lists, arrays, sets, and many custom-defined objects in Java.</p> <p>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 data cannot be modified. Any attempt to modify an immutable object will result in creating a new object with the desired modifications.</p>
<p>6.</p> <p>Ans.</p>	<p>Where exactly is the string constant pool located in the memory?</p> <p>Before Java 8 the string constant pool is stored in the non-heap area of the memory.</p> <p>After java 8 string constant pool and other interned strings are now stored in the native heap.</p> <p>The native heap is a part of the memory managed by the operating system outside of the Java Virtual Machine (JVM) heap. It is where dynamically allocated memory is typically stored.</p>