1. What is a String in Java?

- A String in Java is a sequence of characters. It is an object that represents a sequence of char values. Strings in Java are immutable, meaning once a String object is created, it cannot be changed or modified.

2. Types of String in Java are:

- Immutable String: The standard String class in Java, where any modification creates a new String object.

- Mutable String: Strings that can be modified after creation. Java provides two classes for this: StringBuffer and StringBuilder.

3. In How Many Ways Can You Create String Objects in Java?

- There are two main ways to create String objects in Java:

1. Using String Literal:

   String s1 = "Hello";

   This creates a string in the string constant pool if it doesn't already exist.

2. Using the new Keyword:

   String s2 = new String("Hello");

   This creates a new String object in the heap memory.

4. What is a String Constant Pool?

- The string constant pool is a special memory region where Java stores string literals. When a string literal is created, the JVM checks if it already exists in the pool. If it does, the existing reference is returned; otherwise, the new string is added to the pool.

5. What Do You Mean by Mutable and Immutable Objects?

- Immutable Objects: Objects that cannot be changed after they are created. In Java, String is an example of an immutable object. Any modification to a String creates a new object.

- Mutable Objects: Objects that can be changed after they are created. Examples in Java include

StringBuilder and StringBuffer.

6. Where Exactly is the String Constant Pool Located in the Memory?

- The string constant pool is located in the heap memory. In earlier versions of Java (before Java 7), it was part of the PermGen (Permanent Generation) space, but starting with Java 7, it was moved to the main heap space to improve memory management.