String, StringBuffer, and StringBuilder in Java

1. String (Immutable)

- A String cannot be changed after it is created.

- Any modification creates a new String in memory.

- Best for: fixed text, low modifications, safe use.

Example: String s = "Hello";

2. StringBuilder (Mutable, Not Thread-Safe)

- You can change it without creating a new object.

- Fastest option when working inside a single thread.

- Best for: building text in loops or frequent updates.

Example: StringBuilder sb = new StringBuilder("Hello");

sb.append(" World");

3. StringBuffer (Mutable, Thread-Safe)

- Similar to StringBuilder but synchronized.

- Safe when multiple threads modify the same text.

- Slower than StringBuilder due to thread safety.

Example: StringBuffer sbf = new StringBuffer("Hello");

sbf.append(" World");

Quick Summary: -

String → Immutable, slow with many changes, safe.

- StringBuilder → Mutable, fast, not thread-safe.

- StringBuffer → Mutable, slower, thread-safe.

When to Use:

- Use String when text does not change.

- Use StringBuilder for fast operations in one thread.

- Use StringBuffer when multiple threads modify the same data