StringBuilder vs StringBuffer in Java
Both StringBuilder and StringBuffer are used for **mutable** (modifiable) strings in Java, but they have key differences in terms of **performance** and **synchronization**.

Key Differences

Feature	StringBuffer	StringBuilder
Thread Safety	✅ Synchronized (Safe for multi-threading)	✖ Not synchronized (Faster in single-threaded)
Performance	Slower due to synchronization	Faster as it's not synchronized
Introduced In	Java 1.0	Java 1.5
Use Case	When multiple threads modify the same string	When only a single thread modifies the string

---

Example Code for Both
◈ **StringBuffer (Thread-Safe but Slower)**

```java
public class StringBufferExample {
    public static void main(String[] args) {
        StringBuffer sb = new StringBuffer("Hello");

        sb.append(" World"); // "Hello World"
        sb.insert(5, " Java"); // "Hello Java World"
        sb.replace(6, 10, "C++"); // "Hello C++ World"
        sb.delete(6, 9); // "Hello World"
        sb.reverse(); // "dlroW olleH"

        System.out.println(sb);
    }
}
```

💡 **Use when multiple threads modify the string.**

◈ **StringBuilder (Faster but Not Thread-Safe)**

```java
public class StringBuilderExample {
    public static void main(String[] args) {
        StringBuilder sb = new StringBuilder("Hello");

        sb.append(" World"); // "Hello World"
        sb.insert(5, " Java"); // "Hello Java World"
        sb.replace(6, 10, "C++"); // "Hello C++ World"
        sb.delete(6, 9); // "Hello World"
        sb.reverse(); // "dlroW olleH"

        System.out.println(sb);
    }
}
```

💡 **Use when only a single thread modifies the string for better performance.**

When to Use What?

Scenario	Use
Single-threaded applications	☑ StringBuilder (better performance)
Multi-threaded applications (where multiple threads modify the string)	☑ StringBuffer (ensures thread safety)
Read-Only Strings	☑ Use String instead (immutable and memory-efficient)

🚀 **Conclusion**:

- Use **StringBuilder** for better speed in **single-threaded** environments.
- Use **StringBuffer** only if **thread safety** is required.