

String / StringBuffer / StringBuilder

three siblings, one family, three temperaments — a comparative plate

BY SREENIDHI RAJAKRISHNAN

java.lang • CharSequence • fundamentals

01



String

the carved stone

IMMUTABLE

MUTABILITY	Immutable
THREAD-SAFE	Yes (immutable)
PERFORMANCE	Slow on edits
MEMORY	String Pool
SINCE	JDK 1.0
HASH CACHED	Yes

USE WHEN

Keys, constants, returned values, anything that should never change.

CODE

```
String s = "QA";
s = s + " Engineer";
// new object created
```

02



StringBuffer

the chaperoned ledger

MUTABLE — SAFE

MUTABILITY	Mutable
THREAD-SAFE	Yes (synchronized)
PERFORMANCE	Slower than Builder
MEMORY	Heap
SINCE	JDK 1.0
CAPACITY	Auto-grows

USE WHEN

Mutating shared text across multiple threads where safety beats speed.

CODE

```
StringBuffer b = new StringBuffer("QA");
b.append(" Engineer");
// same object, thread-safe
```

03



StringBuilder

the racing notebook

MUTABLE — FAST

MUTABILITY	Mutable
THREAD-SAFE	No
PERFORMANCE	Fastest of three
MEMORY	Heap
SINCE	JDK 1.5
CAPACITY	Auto-grows

USE WHEN

Single-threaded concatenation, loops, log builders, query stitching.

CODE

```
StringBuilder sb = new StringBuilder("QA");
for (String w : words) sb.append(w);
// fastest, not thread-safe
```

DECISION KEY — read top to bottom; stop at the first YES

01	Will the value never change?	→ use	String
02	Multiple threads will mutate the same text?	→ use	StringBuffer
03	Single-threaded heavy concatenation (loops, logs)?	→ use	StringBuilder