

Java provides two classes for constructing Strings. `StringBuffer` is thread-safe (all methods are synchronized), whereas `StringBuilder` is not. Otherwise, these classes are identical (one can stand-in for the other without code changes).

The following two code examples should generate and return the string "Hello world". They should function correctly in the presence of multiple threads. However, one or both programs *might* contain synchronization errors.

```
// Example A
public class HelloWorldBuilder {
    public String constructString() {
        StringBuffer sb = new StringBuffer();
        sb.append("Hello")
        sb.append(" world");
        sb.append("\n");

        return sb.toString();
    }
}

// Example B
public class HelloWorldBuilder2 {
    private StringBuffer sb = new StringBuffer();

    public String constructString() {
        sb.append("Hello")
        sb.append(" world");
        sb.append("\n");

        String returnValue = sb.toString();

        // reset the buffer before returning
        sb.delete(0, sb.length());

        return returnValue;
    }
}
```

1. Example A is:

- A. Not thread safe.
- B. Not optimal. We can safely replace `StringBuffer` with `StringBuilder`
- C. Just right. The program as written is thread safe and optimal

2. Example B is:

- A. Not thread safe.
- B. Not optimal. We can safely replace `StringBuffer` with `StringBuilder`
- C. Just right. The program as written is thread safe and optimal