**Day 17\_Java Assignment**

**1. Problem Description:**

What is StringBuilder?

What is the purpose of the StringBuilder class because we have a String class also to build string?

**2. My Solution:**

**What is StringBuilder?**

* String builder is a class used to create mutable (modifiable) String. String builder is Asynchronized. This means that multiple thread can call the methods of StringBuilder simultaneously. However, due to its asynchronous nature, StringBuilder is not a thread safe class.
* Since there is no preliminary check for multiple threads, StringBuilder is a lot faster than StringBuffer.

**Example for StringBuilder:**

**package** daily\_assesment;

**public** **class** StringBuilderExample {

**public** **static** **void** main(String[] args) {

StringBuilder sb = **new** StringBuilder("Hello ");

sb.append("Java");

System.***out***.println(sb.toString());

}

}

**Output:**

HelloJava

**What is the purpose of the StringBuilder class because we have a String class also to build string?**

* The String class in Java is immutable, meaning its content cannot be changed. StringBuilder, on the other hand, provides mutable strings, allowing efficient modifications such as concatenation and manipulation. Operations involving string concatenation with the String class can be slower due to the creation of new string objects, whereas StringBuilder offers faster performance

**Example Demonstrating Performance of String and StringBuilder Classes:**

**package** daily\_assesment;

**public** **class** ConcatTest {

**public** **static** **void** main(String[] args) {

// Using String concatenation

**long** startTimeString = System.*nanoTime*();

String resultString = "";

**for** (**int** i = 0; i < 10; i++) {

resultString += "a";

}

**long** endTimeString = System.*nanoTime*();

**long** durationString = endTimeString - startTimeString;

System.***out***.println("Time taken with String concatenation: " + durationString + " nanoseconds");

// Using StringBuilder

**long** startTimeBuilder = System.*nanoTime*();

StringBuilder resultBuilder = **new** StringBuilder();

**for** (**int** i = 0; i < 10; i++) {

resultBuilder.append("a");

}

**long** endTimeBuilder = System.*nanoTime*();

**long** durationBuilder = endTimeBuilder - startTimeBuilder;

System.***out***.println("Time taken with StringBuilder concatenation: " + durationBuilder + " nanoseconds");

}

}

**Output:**

Time taken with String concatenation: 60100 nanoseconds

Time taken with StringBuilder concatenation: 10700 nanoseconds