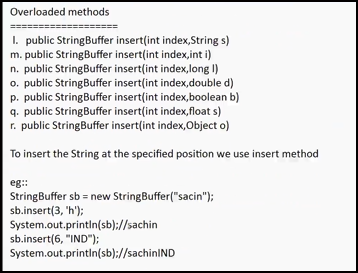
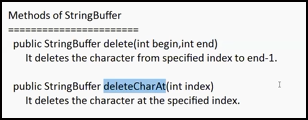
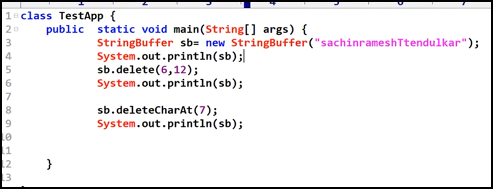


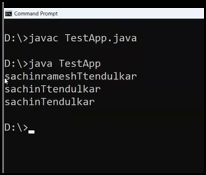
Append adds the Argument at the End.

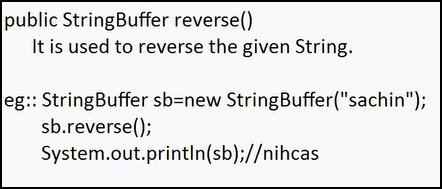


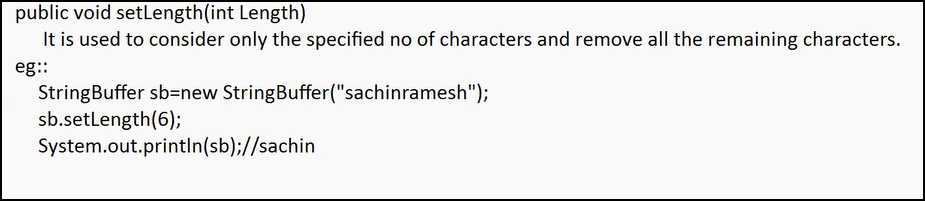
Insert adds the Argument at the Specified index.

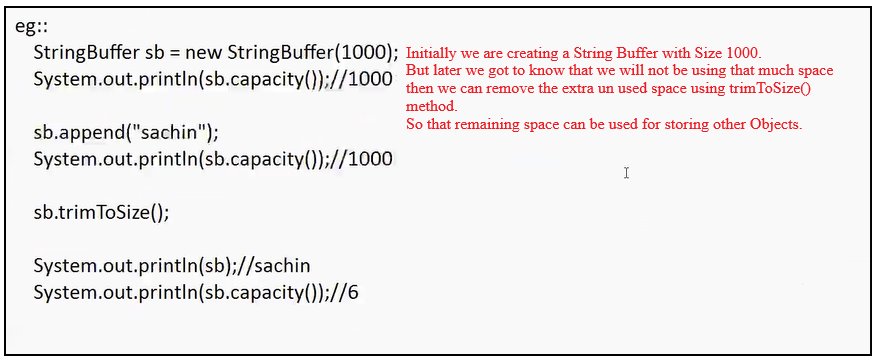


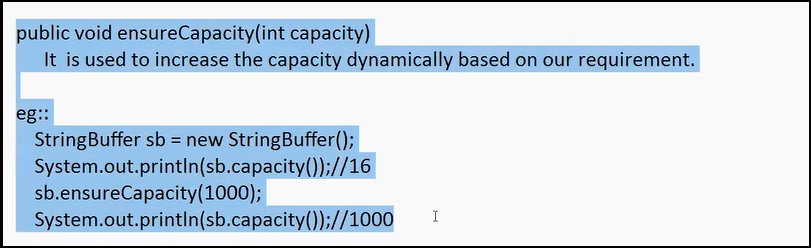


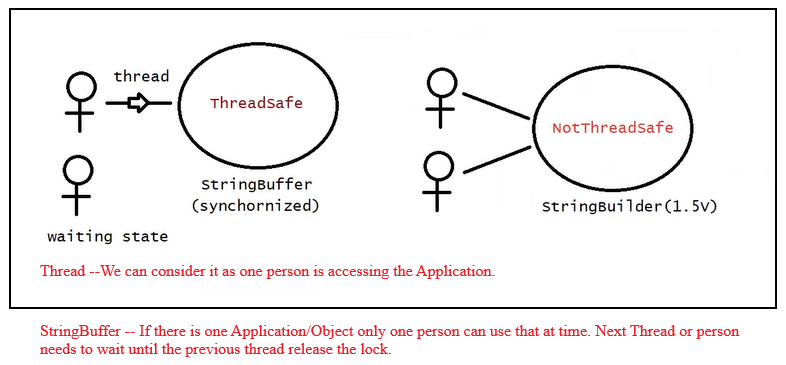


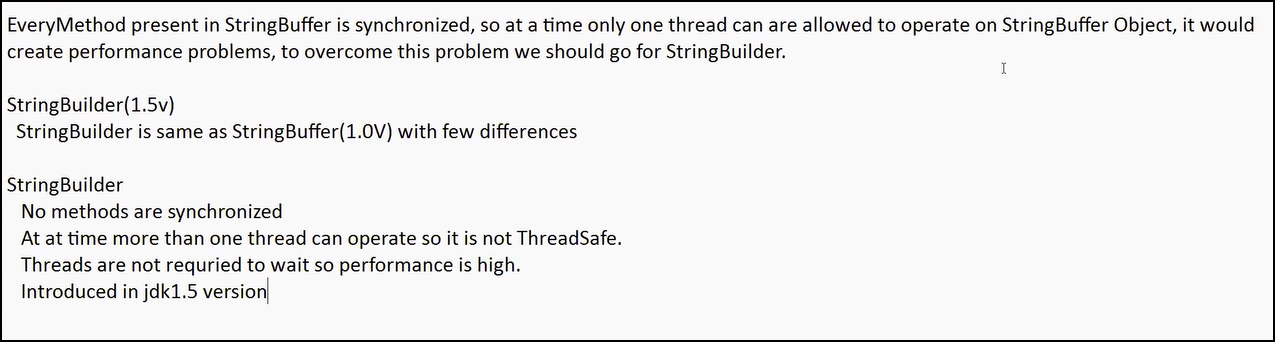


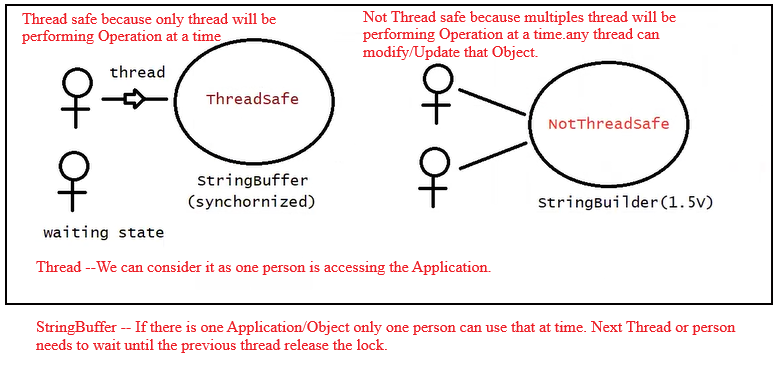












StringBuffer(1.0v) ----Synchornized

Only one Thread can work at time on this.

StringBuilder(1.5v)--Due to the developer's requirement StringBuilder has been introduced. - Not Synchornized

Multiple Threads can work at time on this.

We can create the String in three

String

String Buffer

StringBuilder

when to use which :

String--when Content is fixed

StringBuffer--

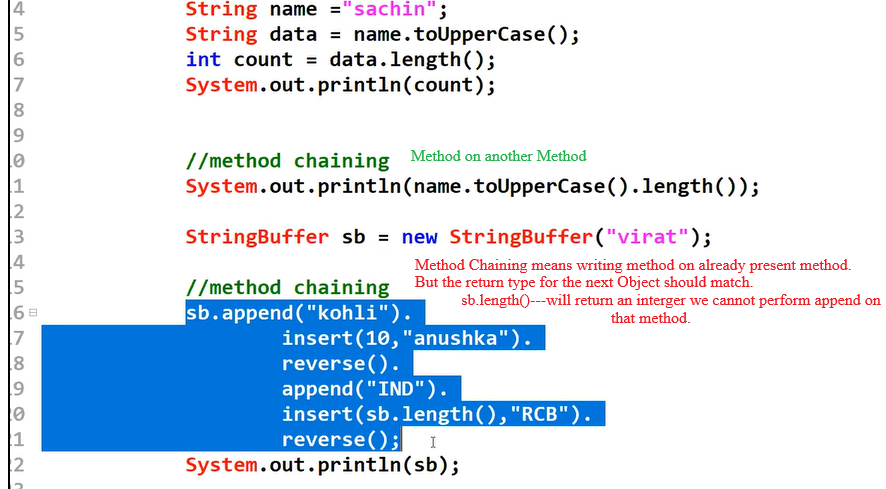
Content can be changed Frequently but less performance.

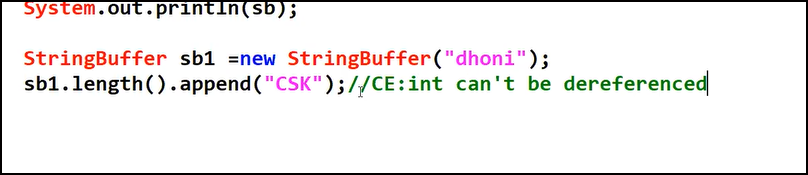
When we want it to be Thread Safe

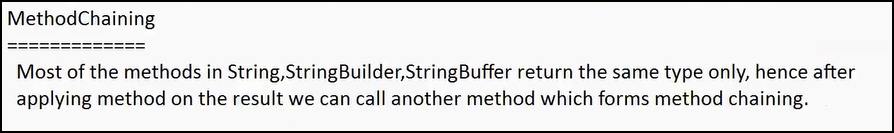
StringBuilder:

Content can be changed Frequently with high performance.

When Thread Safety is not Required.







next() vs nextLine() :

/\*

\* next()---Taken the input until it encounters a Space/Delimeter(/n)

\* ---Keeps the Cursor in the Same Line after taking the input

\* Does not read the newline character.

\*

\* nextLine()

\* ---Taken the input until it encounters a Delimiter(/n) or Enter

\* ---moves the Cursor to the next Line after taking input

\* Reads until the next newline character is encountered, including the newline character.

\*

\* Refer NextVsNextLine code for good understanding regarding next() vs nextLine() difference

\* \*/