**import** java.util.Scanner;

**public class** stringencryption {

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

Scanner scanner = **new** Scanner(System.*in*);

loop:

**while** (**true**) {

System.*out*.println("\n ------------------------------------- " +

"\n Please enter your message to be encrypted (strings only) :)");

String message = scanner.nextLine();

System.*out*.println("\n ----------------------------------- \n" +

"Please confirm that you want to encrypt: " + message + "\n" +

" >> Y/N <<");

**while** (**true**) {

String yesorno = scanner.next().toLowerCase();

scanner.nextLine();

**if** (yesorno.equals("y")) {

String encryptedMessage = *reverseString*(message);

System.*out*.println("\n \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \n" +

"Your encrypted message: " + encryptedMessage);

System.*out*.println("........................ \n this was your original message:" +

"\n" + message);

**break**;

} **else if** (yesorno.equals("n")) {

**continue** loop;

} **else** {

System.*out*.println("I do not understand. Please let me know 'Y' or 'N'");

**continue**;

}

}

scanner.close();

}

}

/\*

String is the return and input type:

input is the parameter that the method accepts --> for the string's "input" that I want to reverse

by using for, from the input.length, from the last string to the first (i=0)

then by using the "reverse function" I can re-generate the input from the last to the first character

\*/

/\*

StringBuilder is commonly used to efficiently construct strings that may be:

1) appending

2) inserting

3) deleting

4) or replacing characters...etc.

In this case it would be replacing characters in the reversed order

It explains that StringBuffer serves a similar purpose;

but Builder is commonly more preferred because it is not synchronized,

meaning - not designed to handle multiple threads (downside is not handling threads "safely")

\*/

**private static** String reverseString(String input) {

StringBuilder reversed = **new** StringBuilder();

**for** (**int** i = input.length() - 1; i>=0; i--){

reversed.append(input.charAt(i));

}

**return** reversed.toString();

}

}