1. Mutable String in Java with example -

Ans - In Java, strings are immutable, which means their values cannot be changed once they are assigned. However, if you need a mutable string-like object, you can use the `StringBuilder` class. `StringBuilder` is mutable, meaning you can modify its contents without creating a new object each time.

Example of `StringBuilder` -

public class Main {

public static void main(String[] args) {

StringBuilder mutableString = new StringBuilder("MutableString");

System.out.println("Original String: " + mutableString);

// Modifying the StringBuilder

mutableString.append("Modified");

System.out.println("Modified String: " + mutableString);

}

}

2. Reverse a String:

Ans –

public class Main {

public static void main(String[] args) {

String input = "PWSKILLS";

StringBuilder reversed = new StringBuilder(input).reverse();

System.out.println("Reversed String: " + reversed);

}

}

3. Reverse a Sentence while Preserving the Position:

Ans -

public class Main {

public static void main(String[] args) {

String input = "Think Twice";

String[] words = input.split(" ");

StringBuilder reversedSentence = new StringBuilder();

for (String word : words) {

StringBuilder reversedWord = new StringBuilder(word).reverse();

reversedSentence.append(reversedWord).append(" ");

}

reversedSentence.deleteCharAt(reversedSentence.length() - 1);

System.out.println("Reversed Sentence: " + reversedSentence);

}

}

4. Sort a String Alphabetically:

Ans -

public class Main {

public static void main(String[] args) {

String input = "cbade";

char[] chars = input.toCharArray();

Arrays.sort(chars);

String sortedString = new String(chars);

System.out.println("Sorted String: " + sortedString);

}

}