**ASSIGNMENT(06-FEB):**

**1.what is mutable string in java explain with an example.**

**Ans:** there is a mutable string class called StringBuilder that allows you to modify the contents of a string without creating a new object. The StringBuilder class is part of the java.lang package and provides methods for appending, inserting, deleting, and modifying characters within a string.

Here's an example that demonstrates the use of StringBuilder**:**

public class MutableStringExample {

public static void main(String[] args) {

StringBuilder sb = new StringBuilder("Hello");

System.out.println("Original StringBuilder: " + sb);

// Append characters to the StringBuilder

sb.append(" World!");

System.out.println("After appending: " + sb);

// Insert characters at a specific position

sb.insert(5, "Java ");

System.out.println("After inserting: " + sb);

// Replace characters within the StringBuilder

sb.replace(0, 5, "Hi");

System.out.println("After replacing: " + sb);

// Delete characters from the StringBuilder

sb.delete(2, 4);

System.out.println("After deleting: " + sb);

// Convert StringBuilder to a String

String finalString = sb.toString();

System.out.println("Final String: " + finalString);

}

}

In the example above, we create a StringBuilder object and initialize it with the value "Hello". Then, we demonstrate various operations that can be performed on the StringBuilder:

Append: We use the append method to concatenate " World!" to the existing string.

Insert: We use the insert method to insert "Java " at index 5.

Replace: We use the replace method to replace the characters from index 0 to 4 with "Hi".

Delete: We use the delete method to remove characters from index 2 to 3.

Convert to String: Finally, we convert the StringBuilder back to a regular immutable String using the toString method.

2**. write a program to reverse a string**

**input“PWSKILLS”**

**out“SLLIKSPW”**

public class StringReversal {

public static String reverseString(String str) {

// Create a StringBuilder object to store the reversed string

StringBuilder reversed = new StringBuilder();

// Iterate through the characters of the string in reverse order

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

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

}

// Convert the StringBuilder to a String and return it

return reversed.toString();

}

public static void main(String[] args) {

String input = "PWSKILLS";

String reversedString = reverseString(input);

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

}

}

3. **Write a program to reverse a sentance while presrving the position**

**input:Think Twice**

**output:“kniht eciwt ”**

public class SentenceReversal {

public static String reverseSentence(String sentence) {

// Split the sentence into words

String[] words = sentence.split("\\s+");

StringBuilder reversedSentence = new StringBuilder();

// Iterate through each word in reverse order

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

String word = words[i];

// Reverse each word

String reversedWord = reverseString(word);

// Append the reversed word to the reversed sentence

reversedSentence.append(reversedWord);

// Append a space unless it is the last word

if (i > 0) {

reversedSentence.append(" ");

}

}

return reversedSentence.toString();

}

public static String reverseString(String str) {

// Convert the string to a character array

char[] charArray = str.toCharArray();

int start = 0;

int end = charArray.length - 1;

// Reverse the characters by swapping positions

while (start < end) {

char temp = charArray[start];

charArray[start] = charArray[end];

charArray[end] = temp;

start++;

end--;

}

return String.valueOf(charArray);

}

public static void main(String[] args) {

String input = "Think Twice";

String reversedSentence = reverseSentence(input);

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

}

}

4. **write a program to sort a string alphabrtically**

**Ans:** public class StringSort {

public static String sortString(String str) {

// Convert the string to a character array

char[] charArray = str.toCharArray();

// Sort the character array using the Arrays.sort method

Arrays.sort(charArray);

// Create a new string from the sorted character array

String sortedString = new String(charArray);

return sortedString;

}

public static void main(String[] args) {

String input = "openai";

String sortedString = sortString(input);

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

}

}