**String Assignment**

Asg ques3

1.What is Mutable String in Java Explain with an example

**Immutable**means unchanging over time or unable to be changed. Whenever we create a string object of the **String class**, it is by default created immutable in nature. If we change the value of the string, the **JVM**creates a new object.

**Mutable** means changing over time or that can be changed. In a **mutable string**, we can change the value of the string and **JVM**doesn’t create a new object. In a **mutable string**, we can change the value of the string in the same object.  
To create a **mutable string in java**, Java has two classes **[StringBuffer](https://javagoal.com/string-in-java/" \l "14" \t "_blank)**and [**StringBuilder**](https://javagoal.com/string-in-java/#10)where the **String class** is used for the **immutable string**

public class MutableStringExample {

public static void main(String[] args) {

StringBuilder mutableString = new StringBuilder("Hello");

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

// Append to the string

mutableString.append(" World!");

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

// Insert into the string

mutableString.insert(5, " Java");

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

// Delete from the string

mutableString.delete(5, 10);

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

// Update a character

mutableString.setCharAt(5, '!');

System.out.println("After updating: " + mutableString);

}

}

2. WAP to reverse a String

Input: “PWSKILLS”

Output: “SLLIKSPW”

This problem can be solved by using reverseString() method and without using reverse method

1. Using reverseString()

**package** Stings;

**public** **class** PracticeReverse {

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

// **TODO** Auto-generated method stub

String str="PWSKILLS";

String reverse="";

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

{

reverse=reverse+str.charAt(i);

}

System.***out***.println("Input:" +str);

System.***out***.println("Output:"+reverse);

}

}

2)By using the reverseString()method

public class ReverseString {

public static void main(String[] args) {

String input = "PWSKILLS";

String reversed = reverseString(input);

System.out.println("Input: " + input);

System.out.println("Output: " + reversed);

}

public static String reverseString(String str) {

StringBuilder reversed = new StringBuilder();

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

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

}

return reversed.toString();

}

}

3. WAP to reverse a sentence while preserving the position

Input:Think Twice

Output:”kniht eciwt”

public class ReverseSentence {

public static void main(String[] args) {

String input = "Think Twice";

String output = reverseSentence(input);

System.out.println("Input: " + input);

System.out.println("Output: " + output);

}

public static String reverseSentence(String sentence) {

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

StringBuilder reversedSentence = new StringBuilder();

for (String word : words) {

StringBuilder reversedWord = new StringBuilder(word);

reversedWord.reverse();

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

}

return reversedSentence.toString().trim();

}

}

4.WAP to sort a String Alphabetically

**package** Stings;

**import** java.util.Arrays;

**public** **class** Sort\_String {

**public** **void** sortString(String s)

{

**char**[] arr= s.toCharArray();

Arrays.*sort*(arr);

System.***out***.print(String.*valueOf*(arr));

}

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

// **TODO** Auto-generated method stub

String str="saurabh";

Sort\_String s =**new** Sort\_String();

s.sortString(str);

}

}

}