# PROGRAMMING USING JAVA WEEK 6 ASSIGNMENT

1. **Create a method which can perform a particular String operation based on the user’s choice. The method should accept the String object and the user’s choice and return the output of the operation. Options are**

# A: Add the String to itself

**B: Replace alternate positions with \***

# C: Remove duplicate characters in the String D:

**Change alternate characters to upper case**

|  |  |
| --- | --- |
| **Method Name** | **changeString** |
| **Method Description** | **Modify the string based on user choice** |
| **Argument** | **String string, char ch** |
| **Return Type** | **String** |
| **Logic** | **Perform the required operation based on the user choice and return the resulting string** |

import java.util.\*; class GFG

{

static String removeDuplicate(char str[], int n) {

// Used as index in the modified string int index = 0;

// Traverse through all characters for (int i = 0; i < n; i++)

{

// Check if str[i] is present before it int j; for (j = 0; j < i; j++)

{

if (str[i] == str[j])

{

break;

}

}

// If not present, then add it to

// result. if (j == i)

{

str[index++] = str[i];

}

}

return String.valueOf(Arrays.copyOf(str, index));

}

// Driver code

public static void main(String[] args)

{

char str[] = "geeksforgeeks".toCharArray(); int n = str.length; System.out.println(removeDuplicate(str, n)); }

# What is the difference between STRING BUILDER AND STRING BUFFER. String Builder :

* + ● StringBuilder is *non-synchronized* i.e. not thread safe. It means two threads can call the methods of StringBuilder simultaneously.
  + ● StringBuilder is *more efficient* than StringBuffer.
  + ● tringBuilder was introduced in Java 1.5

# String Buffer :

* + ● StringBuffer is *synchronized* i.e. thread safe. It means two threads can't call the methods of StringBuffer simultaneously.
  + ● StringBuffer is *less efficient* than StringBuilder.
  + ● StringBuffer was introduced in Java 1.0

# Write a program called Bin2Dec to convert an input binary string into its equivalent decimal number. Your output shall look like:

**Enter a Binary string: 1011**

# The equivalent decimal number for binary "1011" is 11 Enter a Binary string: 1234

**Error: Invalid Binary String "1234"**

import java.util.Scanner; class BinaryToDecimal {

public static void main(String args[]){ Scanner input = new Scanner( System.in ); System.out.print("Enter a binary string: "); String binaryString =input.nextLine();

System.out.println("The equivalent decimal number for binary is "+Integer.parseInt(binaryString,2));

}

# Output :

Enter a Binary string: **1011**

The equivalent decimal number for binary is 11 Enter a Binary string: **1234**

Error: Invalid Binary String