# 5 FEB  Assignment
# String in java

Q(1). WAP (Right a program ) to remove duplicate from String ( Take any string example with duplicate character ).

```
//
//Q(1). Wap (Write a program) to remove duplicate from a string  ( take
any string example with duplicate character )

class RemoveDuplicate {
    public static void main(String[] args) {

      String str1 = "StringInJavaDuplicateValue";
      System.out.println("The given string is: " + str1);
      System.out.println("After removing duplicates characters the new
string is: " + removeDuplicateChars(str1));
    }

    private static String removeDuplicateChars(String sourceStr) {
      char[] arr1 = sourceStr.toCharArray();
      String targetStr = "";
      for (char value : arr1) {
        if (targetStr.indexOf(value) == -1) {
          targetStr += value;
        }
      }
      return targetStr;
    }

  }
```

Q(2). WAP to print duplicate character from the string ?

```
//
//Q(1). Wap (Write a program) to remove duplicate from a string  ( take
any string example with duplicate character )
//  Q(2).  WAP to print duplicates character from the string ?
```

```java
public class DuplicateCharacters {
  public static void main(String[] args) {
    String string1 = "Great responsibility";
    int count;

    // Converts given string into character array
    char string[] = string1.toCharArray();

    System.out.println("Duplicate characters in a given string: ");
    // Counts each character present in the string
    for (int i = 0; i < string.length; i++) {
      count = 1;
      for (int j = i + 1; j < string.length; j++) {
        if (string[i] == string[j] && string[i] != ' ') {
          count++;
          // Set string[j] to 0 to avoid printing visited character
          string[j] = '0';
        }
      }
      // A character is considered as duplicate if count is greater than 1
      if (count > 1 && string[i] != '0')
        System.out.println(string[i]);
    }
  }
}
```

Q(3). WAP to check if “2552” palindrome or not ?

```java
public class palindrome {
    public static void main(String args[]) {
        String str1 = "2552";
        String str2 = "";
        for (int i = str1.length() - 1; i >= 0; i--) {
            str2 = str2 + str1.charAt(i);
        }
        if (str1.equals(str2)) {
            System.out.println("palindrome");
        } else {
```

```
            System.out.println("not palindrome");
        }
    }

}
```

Q(4). WAP to count the number of consonantes , vowel , special character in string ?

```
// Java Program to count vowels, consonant, digits and
// special character in a given string
//Q(4).   WAP to count the number  of consonantes , vowel , special
character in string ?

import java.io.*;

public class countNumbers {

    // Function to count number of vowels, consonant,
    // digitsand special character in a string.
    static void countCharacterType(String str) {
        // Declare the variable vowels, consonant, digit
        // and special characters
        int vowels = 0, consonant = 0, specialChar = 0,
                digit = 0;

        // str.length() function to count number of
        // character in given string.
        for (int i = 0; i < str.length(); i++) {

            char ch = str.charAt(i);

            if ((ch >= 'a' && ch <= 'z') ||
                    (ch >= 'A' && ch <= 'Z')) {

                // To handle upper case letters
                ch = Character.toLowerCase(ch);
```

```
                if (ch == 'a' || ch == 'e' || ch == 'i' ||
                        ch == 'o' || ch == 'u')
                    vowels++;
                else
                    consonant++;
            } else if (ch >= '0' && ch <= '9')
                digit++;
            else
                specialChar++;
        }

        System.out.println("Vowels: " + vowels);
        System.out.println("Consonant: " + consonant);
        System.out.println("Digit: " + digit);
        System.out.println("Special Character: " + specialChar);
    }

    // Driver function.
    static public void main(String[] args) {
        String str = "sumit dongre 1234";

        countCharacterType(str);
    }
}

// This code is contributed by vt_m.
```

Q(5). WAP to implement Anagram checking least inbuilt method being used ?

```
import java.util.Arrays;

public class anagram {
    public static void main(String args[]) {

        String str1 = "School Master";
        String str2 = "The Classroom";
```

```
        str1 = str1.replace("", "");
        str2 = str2.replace("", "");

        str1 = str1.toLowerCase();
        str2 = str2.toLowerCase();
        char[] ar1 = str1.toCharArray();
        char[] ar2 = str2.toCharArray();

        Arrays.sort(ar1);
        Arrays.sort(ar2);

        if (Arrays.equals(ar1, ar2)) {
            System.out.println("it is Anagram");
        } else {
            System.out.println("it is not Anagram");
        }

    }
}
```

Q(6). WAP to implement Pangram checking least inbuilt method being used ?

```
import java.util.Arrays;
import java.io.*;

//Q(6).    WAP to implement Pangram checking least inbuilt method being
used ?

public class Pangram {

    // Java Program to illustrate Pangram

    public static boolean checkPangram(String str) {

        boolean[] mark = new boolean[26];

        int index = 0;

        for (int i = 0; i < str.length(); i++) {
```

```
            if ('A' <= str.charAt(i)
                    && str.charAt(i) <= 'Z')
                index = str.charAt(i) - 'A';

            else if ('a' <= str.charAt(i)
                    && str.charAt(i) <= 'z')

                index = str.charAt(i) - 'a';

            else
                continue;
            mark[index] = true;
        }

        for (int i = 0; i <= 25; i++)
            if (mark[i] == false)
                return (false);

        return (true);
    }

    public static void main(String[] args) {
        String str = "The quick brown fox jumps over the lazy dog";

        if (checkPangram(str) == true)
            System.out.print(str + " \nis a pangram.");
        else
            System.out.print(str + " \nis not a pangram.");
    }
}
```

Q(7). WAP to find if String contains all unique characters ?

```
// Java program to illustrate string with
// unique characters using brute force technique
import java.util.*;
```

```
class uniq12 {
    boolean uniqueCharacters(String str) {
        // If at any time we encounter 2 same
        // characters, return false
        for (int i = 0; i < str.length(); i++)
            for (int j = i + 1; j < str.length(); j++)
                if (str.charAt(i) == str.charAt(j))
                    return false;

        // If no duplicate characters encountered,
        // return true
        return true;
    }

    public static void main(String args[]) {
        uniq12 obj = new uniq12();
        String input = "GeeksforGeeks";

        if (obj.uniqueCharacters(input))
            System.out.println("The String " + input + " has all unique
characters");
        else
            System.out.println("The String " + input + " has duplicate
characters");
    }
}
```

Q(8). WAP to find the maximum occurring character in string ?

```
// Java program to output the maximum occurring character
// in a string

//Q(8).  WAP to find the maximum occurring character in string ?

public class MaxOcc12 {
    static final int ASCII_SIZE = 256;

    static char getMaxOccurringChar(String str) {
```

```
        int count[] = new int[ASCII_SIZE];

        int len = str.length();
        for (int i = 0; i < len; i++)
            count[str.charAt(i)]++;

        int max = -1; // Initialize max count
        char result = ' '; // Initialize result

        for (int i = 0; i < len; i++) {
            if (max < count[str.charAt(i)]) {
                max = count[str.charAt(i)];
                result = str.charAt(i);
            }
        }

        return result;
    }

    // Driver Method
    public static void main(String[] args) {
        String str = "sample string";
        System.out.println("Max occurring character is "
                + getMaxOccurringChar(str));
    }
}
```