**Module no.2**

**Oop with java**

**LabAssignment no. 4**

**Q.1 Write a program which takes a string and prints the number of vowels, consonants(non-vowels). numbers and other character**

**Input: "Hello world 37 1!" Output: Vowels: 3 Consonants: 7 Numbers: 2 Others: 6**

import java.io.\*;

import java.util.Scanner;

public class Sol{

static void countCharacterType(String str)

{

int vowels = 0, consonant = 0, specialChar = 0,

digit = 0;

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

char ch = str.charAt(i);

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

(ch >= 'A' && ch <= 'Z') ) {

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);

}

public static void main (String[] args)

{

//String str = "I am learning java1.8";

String str;

System.out.println("Enter the String");

Scanner sc=new Scanner(System.in);

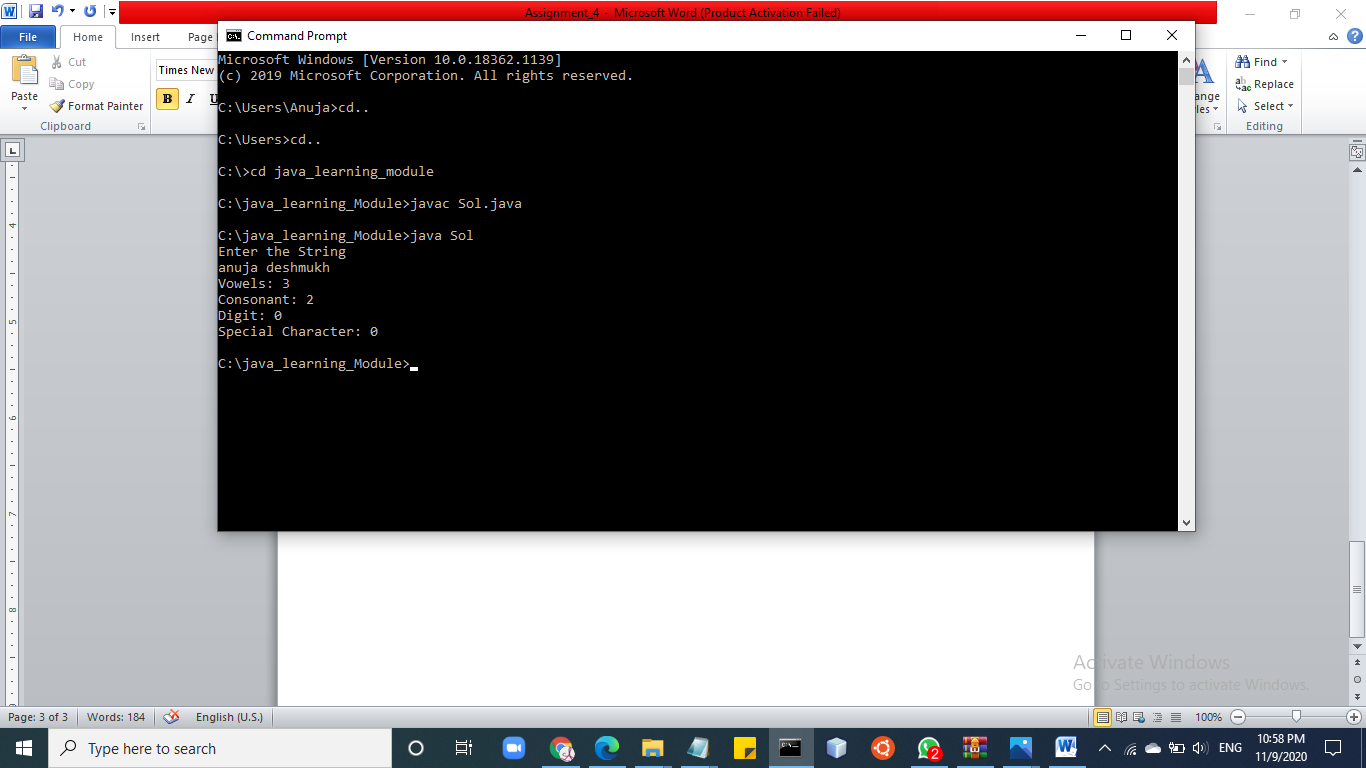
str=sc.next();

countCharacterType(str);

}

}

**//Output**



**Q2. Given a string which contains numbers from 0 to 9 expressed as words, output a string which contains all numbers incremented by 1**

Input : There are three bugs and nine features

Output : There are four bugs and ten features

**Q3. Write a function to replace multiple consecutive occurrences of characters with a single character**

**Input : abccddde Output : abcde**

**Input : aabbbbaaa Output : aba**

class Sol{

public static void main(String args[]){

String str ="###aabc";

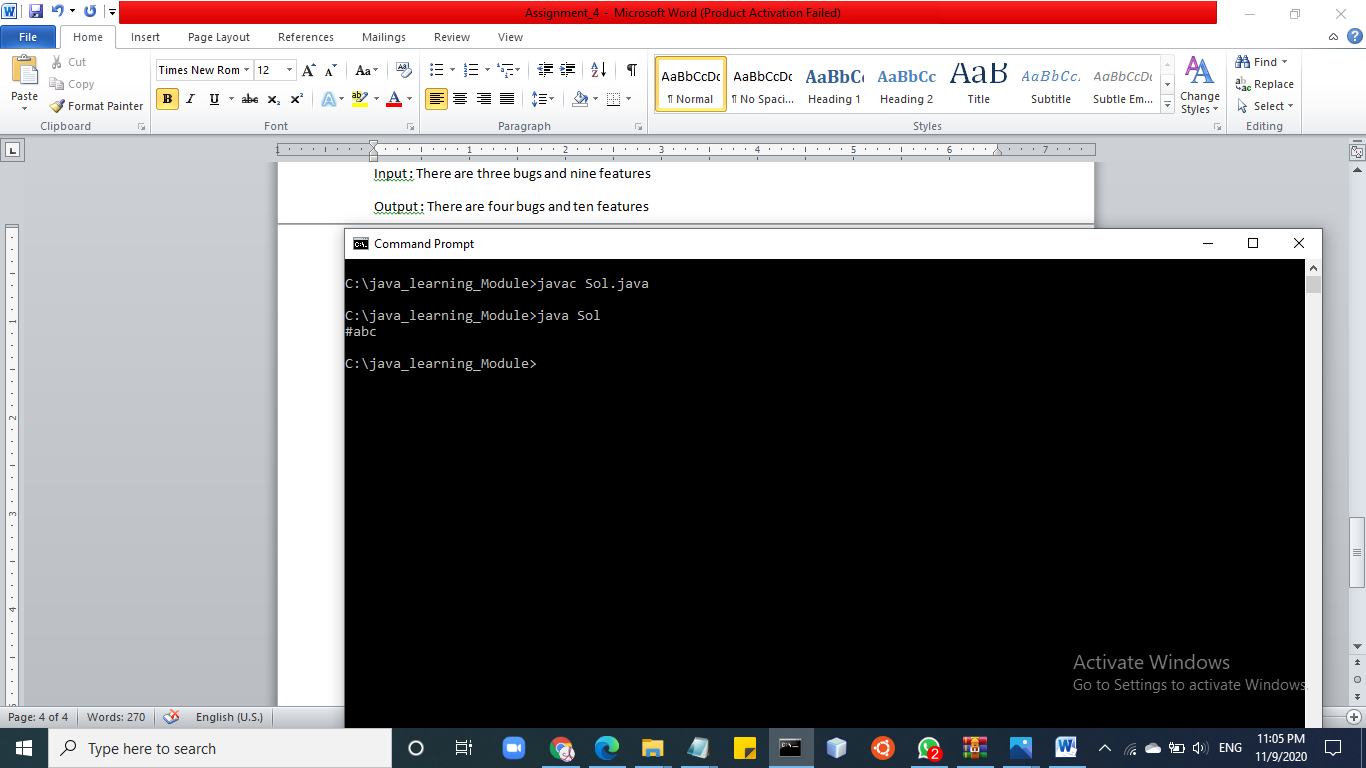
str = str.replaceAll("(.)\\1+","$1");

System.out.println(str);

}

}

**//Output**



**Q4. Write a program to reverse a string. Input : "hello" Output : "olleh" Input : "Hello World" Output : "dlrow olleh"**

import java.util.\*;

class Sol

{

public static void main(String args[])

{

String original, reverse = "";

Scanner in = new Scanner(System.in);

System.out.println("Enter a string to reverse");

original = in.nextLine();

int length = original.length();

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

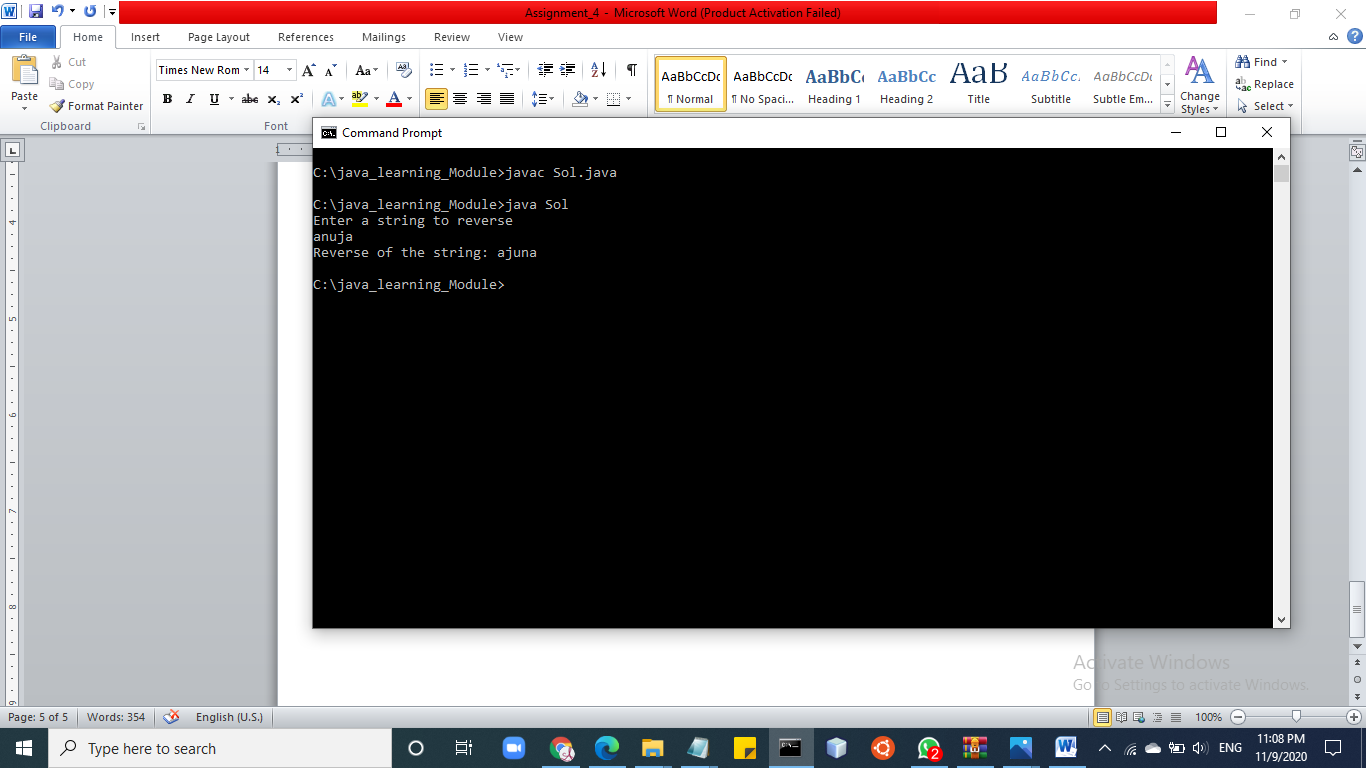
reverse = reverse + original.charAt(i);

System.out.println("Reverse of the string: " + reverse);

}

}

//Output



**Q5. Write a program which accepts two string and prints all unique characters which are common in both stirngs.**

**Input 1 : "hello world" Input 2 : "lot of work" output : 'l' , 'o' , 'w' , 'r'**

import java.io.\*;

import java.util.\*;

public class Sol

{

static final int MAX\_CHAR = 26;

static void printCommon(String s1, String s2)

{

int[] a1 = new int[MAX\_CHAR];

int[] a2 = new int[MAX\_CHAR];

int length1 = s1.length();

int length2 = s2.length();

for (int i = 0 ; i < length1 ; i++)

a1[s1.charAt(i) - 'a'] += 1;

for (int i = 0 ; i < length2 ; i++)

a2[s2.charAt(i) - 'a'] += 1;

for (int i = 0 ; i < MAX\_CHAR ; i++)

{

if (a1[i] != 0 && a2[i] != 0)

{

for (int j = 0 ; j < Math.min(a1[i], a2[i]) ; j++)

System.out.print(((char)(i + 'a')));

}

}

}

public static void main(String[] args) throws IOException

{

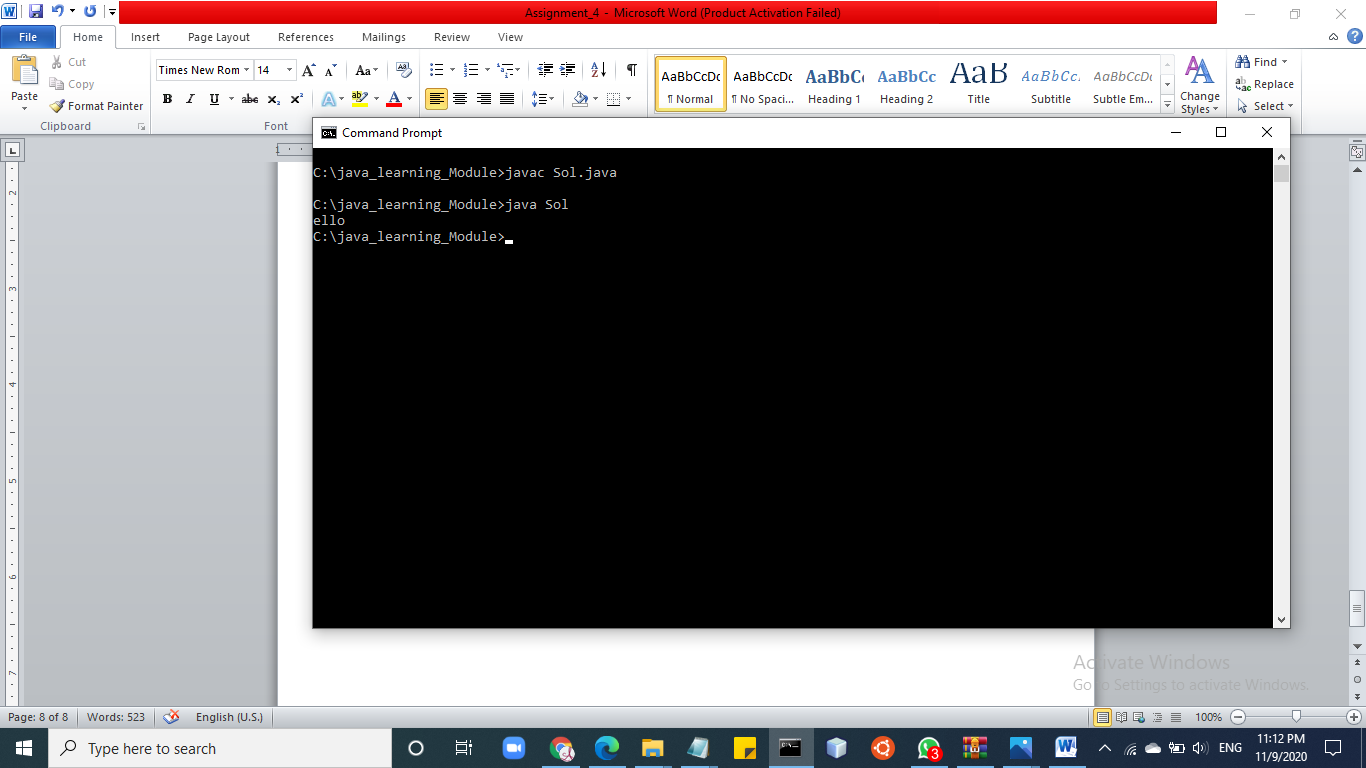
String s1 = "hello", s2 = "oneandall";

printCommon(s1, s2);

}

}

//Output



**Q6. Given a string , output another string where the case of the characters is reversed. Input : "Hello World" Output : "hELLO wORLD"**

class Sol{

static void convertOpposite(StringBuffer str)

{

int ln = str.length();

for (int i=0; i<ln; i++)

{

Character c = str.charAt(i);

if (Character.isLowerCase(c))

str.replace(i, i+1, Character.toUpperCase(c)+"");

else

str.replace(i, i+1, Character.toLowerCase(c)+"");

}

}

public static void main(String[] args)

{

StringBuffer str = new StringBuffer("Hello World");

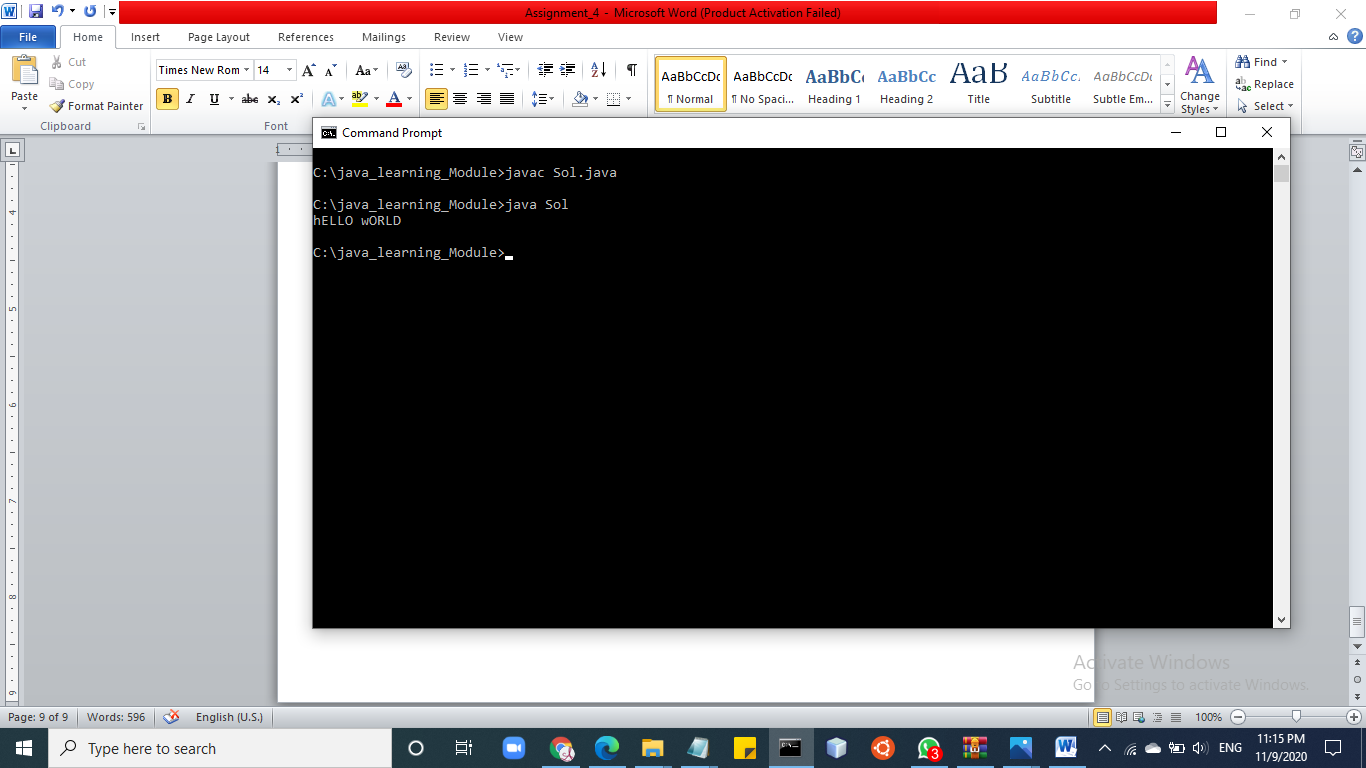
convertOpposite(str);

System.out.println(str);

}

}

**//Output**



**Q7. Given a string, check whether it is a formed by concatenating the same string 3 times. Input: "abcabcabc", Output: true (abc is repeated 3 times) Input: "abcdabcdabcd", Output: true (abcd is repeated 3 times) Input: "andandan', Output: false**