|  |  |
| --- | --- |
| **Name - Rachit Mody** | **SAP ID - 60004220248** |

**Experiment No - 1**

**AIM: To implement Java Program Structures and Simple Programs.**

1.1 WAP to display Hello World message on screen.

**THEORY:**

In Java, System.out.println() is used to print a statement which has been passed in its argument. There are 2 printing statements in Java, the first being System.out.print() which prints the argument passed through it on the same line and the second being System.out.println() which is similar to System.out.print() method except that it moves the cursor to the next line after printing the result.

**CODE:**

class hello\_world{

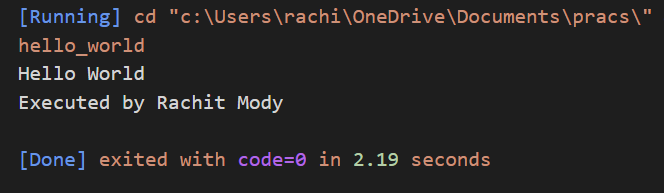
public static void main(String args[]){

System.out.println("Hello World");

System.out.println("Executed by Rachit Mody");

}

}

**OUTPUT:**

1.2 Write a Java program that reads a positive integer from the command line and count the number of digits the number (less than ten billion) has.

**THEORY**

The while loop in Java is a control flow statement that allows code to be executed repeatedly based on a given boolean condition. The loop goes on until the boolean condition turns false. When the number of iterations in not known to the user, they can use the while loop.

**CODE:**

import java.util.Scanner;

public class no\_of\_digits {

public static void main(String args[]){

Scanner sc = new Scanner(System.in);

System.out.println("Enter a number:");

long no = sc.nextLong();

int count = 0;

while(no>0){

no= no/10;

count++;

}

System.out.println("Number of digits: "+count);

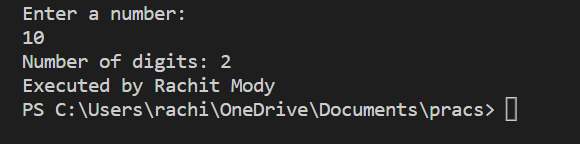
System.out.println("Executed by Rachit Mody");

sc.close();

}

}

**OUTPUT:**



**CONCLUSION:**

Hereby, we have performed a simple program using Java

**Experiment No - 2**

**AIM: To implement Java control Statements and loops.**

2.1 WAP to find roots of Quadratic equation. Take care of imaginary values.

**THEORY:**

Java has the following conditional statements:

• If to specify a block of code to be executed, if a specified condition is true

• else to specify a block of code to be executed, if the same condition is false

• else if to specify a new condition to test, if the first condition is false

**CODE:**

import java.util.Scanner;

public class roots {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

System.out.println("Enter a b and c");

double a = input.nextInt();

double b = input.nextInt();

double c = input.nextInt();

double deter = b \* b - 4 \* a \* c;

if (deter > 0) {

double deter\_root = Math.sqrt(deter);

System.out.println("Root 1=" + (-b + deter\_root) / (2 \* a));

System.out.println("Root 2=" + (-b - deter\_root) / (2 \* a));

} else if (deter == 0) {

double deter\_root = Math.sqrt(deter);

System.out.println("Root 1=Root 2" + (-b + deter\_root) /

(2 \* a));

} else {

double deter\_root = Math.sqrt(-deter);

double real = -b / (2 \* a);

double img = deter\_root / (2 \* a);

System.out.println("Root 1=" + real+"+"+img+"i");

System.out.println("Root 1=" + real+"-"+img+"i");

}

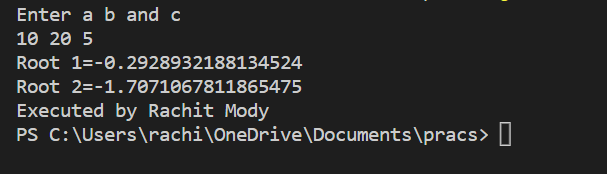
input.close();

System.out.println("Executed by Rachit Mody");

}

}

**OUTPUT:**



2.2 Write a menu driven program using switch case to perform mathematical operations

**THEORY:**

Java has another conditional statement called switch. We use switch to specify many alternative blocks of code to be executed. The value for a case must be a constant or a literal. Variables are not allowed. The break statement is used inside the switch to terminate a statement sequence.

**CODE:**

import java.util.Scanner;

public class mathops

{

public static void main(String args[])

{

int a,b,ch;

Scanner sc = new Scanner(System.in);

System.out.println("1.Addition\n 2.subtraction\n 3.Multiplication\n 4.Modulus\n5.Division\n");

System.out.println("Enter the values of a&b: ");

a = sc.nextInt();

b = sc.nextInt();

System.out.println("Enter Your Choice: ");

ch=sc.nextInt();

switch(ch)

{

case 1:

int result=a+b;

System.out.println("sum of A and B is "+ result);

break;

case 2:

int result1=a-b;

System.out.println("Subtraction of A and B is" +result1);

break;

case 3:

int result2=a\*b;

System.out.println("Multiplication of A and B is" +result2);

break;

case 4:

int result3=a%b;

System.out.println("Modulud of A and B is" +result3);

break;

case 5:

int result4=a/b;

System.out.println("Divisionof A and B is" +result4);

break;

default:

System.out.println("Invalid choice Please try again: ");

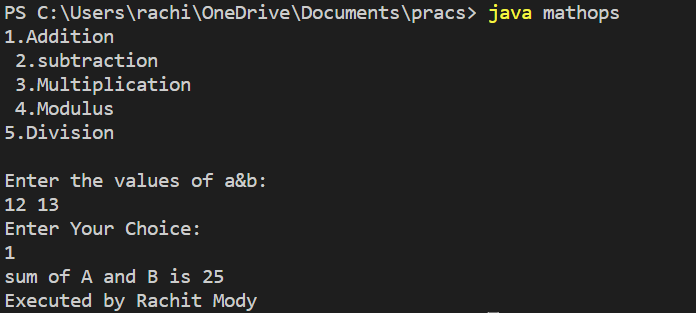
}

System.out.println("Executed by Rachit Mody");

}

}

**OUTPUT:**



2.3 WAP to display odd number from given range.

**THEORY:**

In Java, for loop is used to iterate a part of the program several times. If a user knows the number of iterations, then it is recommended to use a for loop. The for statement consumes the initialization, condition and increment/decrement in one line thereby providing a shorter, easy to debug structure of looping.

**CODE:**

import java.util.\*;

public class oddandprime{

public static void main(String args[])

{

Scanner sc = new Scanner(System.in);

int x;

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

x = sc.nextInt();

System.out.println("The odd number between 1 to "+x +"is");

for(int i=1;i<=x;i++){

if(i%2!=0){

System.out.print(i+" ");

}

}

System.out.println("The prime number between 1 to "+x+"is");

int i,j,chk;

for(i=2; i<=x; i++)

{

chk = 0;

for(j=2; j<i; j++)

{

if(i%j==0)

{

chk++;

break;

}

}

if(chk==0)

System.out.print(i+" ");

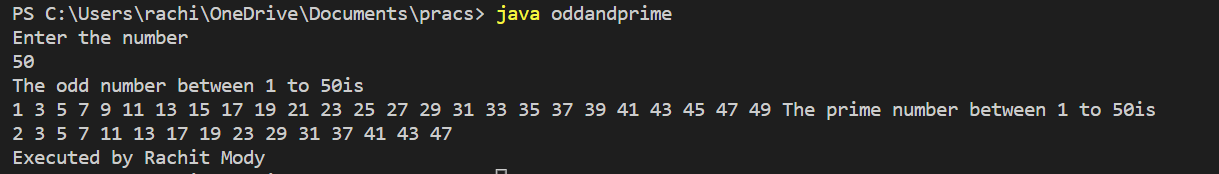
}

System.out.println("\nExecuted by Rachit Mody");

}

}

**OUTPUT:**



2.4 WAP to display default value of primitive data types.

**THEORY:**

Primitive data types are the building blocks of data manipulation. These are the most basic data types available in Java language. The eight primitives defined in Java are int, byte, short,long, float, double, boolean, and char – those aren't considered objects and represent raw values.

**CODE:**

import java.util.\*;

class defaultvalue

{

int i;

float f;

double d;

long l;

boolean bl;

short s;

byte b;

char ch;

public static void main(String args[])

{

defaultvalue sc = new defaultvalue();

System.out.println("The Deafult Value of int is: "+sc.i);

System.out.println("The Deafult Value of float is: "+sc.f);

System.out.println("The Deafult Value of double is: "+sc.d);

System.out.println("The Deafult Value of long is: "+sc.l);

System.out.println("The Deafult Value of boolean is: "+sc.bl);

System.out.println("The Deafult Value of short is: "+sc.s);

System.out.println("The Deafult Value of byte is: "+sc.b);

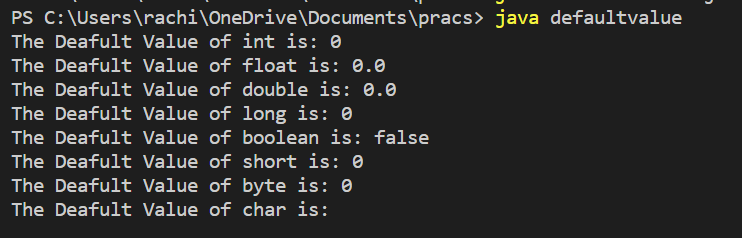
System.out.println("The Deafult Value of char is: "+sc.ch);

System.out.println("\nExecuted by Rachit Mody");

}

}

**OUTPUT:**



2.5 WAP to display the following patterns:

1

2 1

1 2 3

4 3 2 1

1 2 3 4 5

6 5 4 3 2 1

1 2 3 4 5 6 7

**THEORY:**

If a loop exists inside the body of another loop, it's called a nested loop. That is why nested loops are also called as “loop inside loop“.

**CODE:**

import java.util.\*;

public class pattern1{

public static void main(String args[])

{

Scanner sc = new Scanner(System.in);

int x;

System.out.println("Enter the number of rows");

x = sc.nextInt();

for(int i=1;i<=x;i++){

if(i%2 !=0){

for(int j=1;j<=i;j++){

System.out.print(j+" ");

}

}

else{

for(int j=i;j>=1;j--){

System.out.print(j+" ");

}

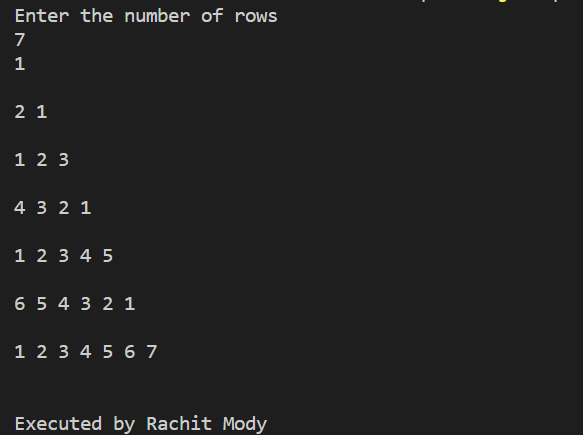
}

System.out.println("\n");

}

System.out.println("\nExecuted by Rachit Mody");

} }

**OUTPUT:**

A

C B

F E D

J I H G

**CODE:**

import java.util.\*;

public class pattern2{

public static void main(String args[])

{

Scanner sc = new Scanner(System.in);

int num,temp=65,alpha,temp1=1;

num = sc.nextInt();

int count = num -1;

for(int i =1;i<=num;i++){

for(int k =1;k<=count;k++)

{

System.out.print(" ");

}

alpha = temp;

for(int j=1;j<=i;j++){

System.out.print((char)(alpha));

alpha--;

}

temp = temp + (++temp1);

System.out.println("\n");

count--;

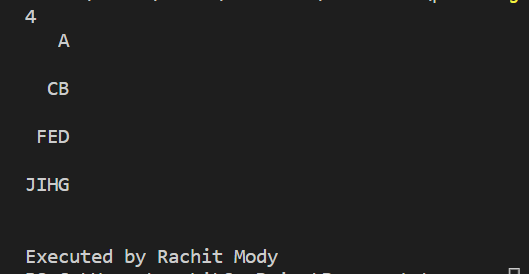
}

System.out.println("\nExecuted by Rachit Mody");

}

}

**OUTPUT:**



**CONCLUSION:**

Thus, we implemented Java control Statements and loops.

**Experiment No - 3**

3.1 WAP to find whether the entered 4 digit number is vampire or not. Combination of digits from this number forms 2 digit number. When they are multiplied by each other we get the original number.

**THEORY:**

An array is a container object that holds a fixed number of values of a single type. The length of an array is established during the creation of the array.It is used to store multiple values in a single variable instead of declaring multiple variables fro each value.

**CODE:**

import java.util.\*;

public class vam {

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

int size=4;

int num,nums[],digits[];

nums=new int[size];

digits=new int[size];

num=sc.nextInt();

for(int i=0;i<4;i++)

{

digits[i]=(num/(int)Math.pow(10,size-i-1))%10;

}

System.out.println();

nums[0]=digits[0]\*10+digits[1];

nums[1]=digits[1]\*10+digits[0];

nums[2]=digits[2]\*10+digits[3];

nums[3]=digits[3]\*10+digits[2];

if(nums[0]\*nums[2]==num||nums[0]\*nums[3]==num||nums[1]\*nums[2]==num||nums[0

]\*nums[3]==num)

{

System.out.println(num+" is vampire");

}

else

{

System.out.println(num+" is not vampire");

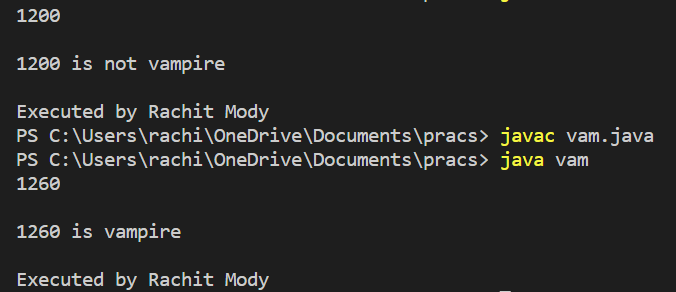
}

System.out.println("\nExecuted by Rachit Mody");

}

}

**OUTPUT:**



3.2 WAP to display the following using irregular arrays

1

2 3

4 5 6

**THEORY:**

Irregular arrays or jagged arrays is a group of arrays where each array can be of different sizes. We can create a 2D array with variable number of columns in each row.

**CODE:**

public class JaggedArray {

public static void main(String[] args) {

int arr[][] = new int[3][];

arr[0] = new int[1];

arr[1] = new int[2];

arr[2] = new int[3];

int count = 1;

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

for (int j = 0; j < arr[i].length; j++) {

arr[i][j] = count++;

}

}

System.out.println("Irregular Array");

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

for (int j = 0; j < arr[i].length; j++) {

System.out.print(arr[i][j] + " ");

}

System.out.println();

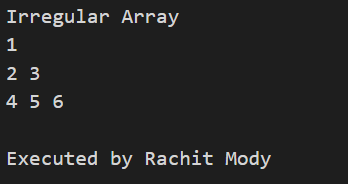
}

System.out.println("\nExecuted by Rachit Mody");

}

}

**OUTPUT:**



3.3 Write a program that queries a user for the no : of rows and columns representing students and their marks .Reads data row by row and displays the data in tabular form along with the row totals , columns total and grand total.

**THEORY:**

Multidimensional arrays is an array of arrays. Each element of a multidimensional array is an array itself. The total number of elements that can be stored in a multidimensional array can be calculated by multiplying the size of all the dimensions.

**CODE:**

import java.util.\*;

public class Main1 {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Enter rows of matrix : ");

int r = sc.nextInt();

System.out.print("Enter colums of matrix : ");

int c = sc.nextInt();

int matrix[][] = new int[r][c];

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

for (int j = 0; j < c; j++)

matrix[i][j] = sc.nextInt();

int sumOfRow[] = new int[r];

Arrays.fill(sumOfRow, 0);

int sumOfCol[] = new int[c];

Arrays.fill(sumOfCol, 0);

int grandTotal = 0;

for (int i = 0; i < r; i++) {

for (int j = 0; j < c; j++) {

sumOfRow[i] += matrix[i][j];

sumOfCol[j] += matrix[i][j];

}

grandTotal += sumOfRow[i];

}

for (int i = 0; i < r; i++) {

for (int j = 0; j < c; j++) {

System.out.print(matrix[i][j] + "\t");

}

System.out.print("| " + sumOfRow[i] + " \n");

}

for (int j = 0; j < c; j++) {

System.out.print("----\t");

}

System.out.println("----");

for (int j = 0; j < c; j++) {

System.out.print(sumOfCol[j] + "\t");

}

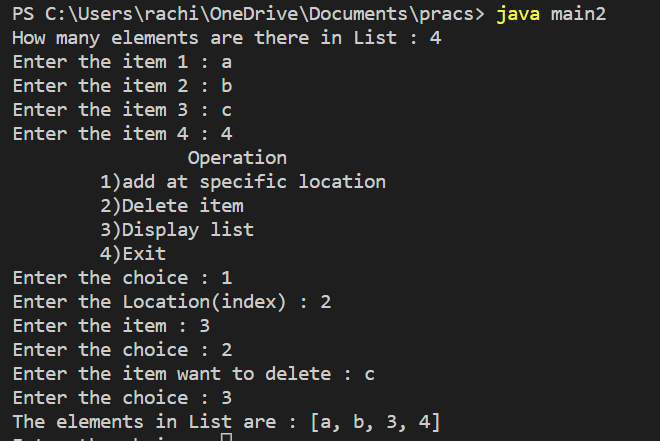
System.out.print("| " + grandTotal + "\t");

System.out.println("\nExecuted by Rachit Mody");

}

}

**OUTPUT:**



**CONCLUSION:** Thus, we implemented programs on Array.

**Experiment No - 4**

**AIM: To implement Vectors**

4.1 WAP that accepts a shopping list of items and performs the following operations: Add an item at a specified location, delete an item in the list, and print the contents of the vector

**THEORY:**

The Vector class implements a growable array of objects i.e. it can grow and shrink as per the requirements of the user. Unlike array, we can store nnumber of elements in it as there is no size limit.

**CODE:**

import java.util.\*;

public class main2 {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("How many elements are there in List : ");

int n = sc.nextInt();

Vector v = new Vector(n, 3);

for (int i = 0; i < n; i++) {

System.out.printf("Enter the item %d : ", i + 1);

String str = sc.next();

v.addElement(str);

}

System.out.println("\t\tOperation");

System.out.println("\t1)add at specific location");

System.out.println("\t2)Delete item");

System.out.println("\t3)Display list");

System.out.println("\t4)Exit");

int choice = 1;

while (choice != 4) {

System.out.print("Enter the choice : ");

choice = sc.nextInt();

switch (choice) {

case 1:

System.out.print("Enter the Location(index) : ");

int l = sc.nextInt();

System.out.print("Enter the item : ");

String str = sc.next();

v.add(l, str);

break;

case 2:

System.out.print("Enter the item want to delete : ");

String item = sc.next();

v.remove(item);

break;

case 3:

System.out.print("The elements in List are : " + v + "\n");

break;

case 4:

return;

default:

System.out.print("wrong choice ");

}

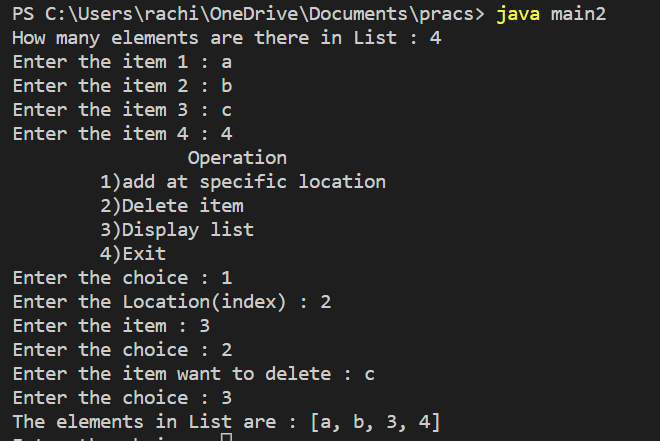
}

System.out.println("\nExecuted by Rachit Mody");

}

}

**OUTPUT:**



4.2 Write a java program to find the frequency of an element in the given Vector array.

**THEORY:**

Vectors in java supports constructors and methods.

Some of the constructors are

* vector()-It constructs an empty vector with the default size as 10.
* vector(int initialCapacity)- It constructs an empty vector with the specified initial capacity

and with its capacity increment equal to zero.

* vector(int initialCapacity, int capacityIncrement)- It constructs an empty vector with the

specified initial capacity and capacity increment.

* add()-It is used to append the specified element in the given vector.
* size()-It is used to get the number of components in the given vector.
* remove()-It is used to remove the specified element from the vector. If the vector does not

contain the element, it is unchanged.

* get()-It is used to get an element at the specified position in the vector.

**CODE:**

import java.util.\*;

class main3 {

public static void main(String args[]) {

int i, n, x, count = 0;

Scanner scan = new Scanner(System.in);

System.out.print("Enter the size of the vector array:");

n = scan.nextInt();

System.out.println("Enter " + n + " elements:");

Vector v = new Vector();

for (i = 0; i < n; i++) {

System.out.print("Enter the element for position " + (i + 1) + ":");

v.addElement(scan.nextInt());

}

System.out.println("Vector array:" + v);

System.out.print("Enter the element whose frequency to be known:");

x = scan.nextInt();

int startpos=0,newpos;

while(startpos<v.size())

{

newpos=v.indexOf(x,startpos);

if(newpos==-1)

{

break;

}

count++;

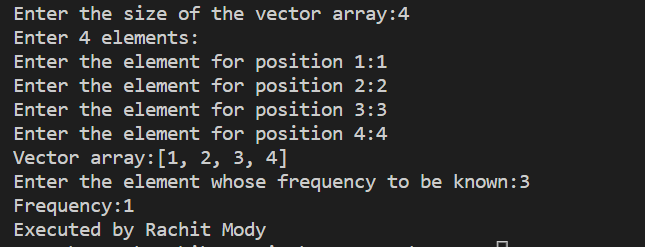
startpos=newpos+1;

}

System.out.print("Frequency:" + count);

System.out.println("\nExecuted by Rachit Mody");}}

**OUTPUT:**



**CONCLUSION:** Thus, we implemented Vectors.

**Experiment No - 5**

**AIM: To implement Strings.**

5.1 WAP to check if 2 strings are Meta strings or not. Meta strings are the strings which can be made equal by exactly one swap in any of the strings. Equal string are not considered here as Meta strings.

**THEORY:**

String is basically an object that represents sequence of char values.

An array of characters works same as Java string.

Example: char[] ch={'R','a','c','h','i',’t’};

String s=new String(ch);

Is the same as

String s="Rachit";

**CODE:**

import java.util.\*;

public class main4 {

static boolean isMeta(String word1,String word2)

{

int wrong\_count=0,wrong\_pos[];

wrong\_pos=new int[2];

if(word1.length()!=word2.length())

return false;

for(int i=0;i<word1.length();i++)

{

if(word1.charAt(i)!=word2.charAt(i))

{

wrong\_count++;

if(wrong\_count>2)

return false;

wrong\_pos[wrong\_count-1]=i;

}

}

if(wrong\_count==2)

{

if(word1.charAt(wrong\_pos[0])==word2.charAt(wrong\_pos[1])&&word1.charAt(wrong\_pos[1])= =word2.charAt(wrong\_pos[0]))

return true;

}

return false;

}

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

String word1,word2;

word1=sc.next();

word2=sc.next();

if(isMeta(word1, word2))

{

System.out.println("Strings are meta");

}

else

{

System.out.println("Strings are not meta");

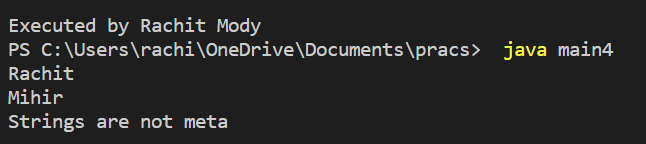
}

System.out.println("\nExecuted by Rachit Mody");

}

}

**OUTPUT:**



5.2 Write a java program to count number of alphabets, digits, special symbols, blank spaces and words from the given sentence. Also count number of vowels and consonants.

**THEORY:**

Java string class has many useful methods to perform operations on sequence of char values. Some of the

are listed below:

* int length()-It returns string length
* boolean contains(CharSequence s)- It returns true or false after matching the sequence of char

value.

* String toUpperCase()-Converts all of the characters in this String to upper case using the rules of

the default locale.

* String toLowerCase()-Converts all of the characters in this String to lowercase using the rules of

the default locale.

**CODE:**

import java.util.\*;

class main5 {

public static void main(String args[]) {

int v = 0;

int c = 0;

String str;

Scanner scan = new Scanner(System.in);

System.out.println("Enter a sentence:");

str = scan.nextLine();

count(str);

word(str);

str = str.toLowerCase();

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

char ch = str.charAt(i);

if (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u') {

v++;

} else if ((ch >= 'a' && ch <= 'z')) {

c++;

}

}

System.out.println("No.of vowels:" + v);

System.out.println("No.of consonants:" + c);

}

public static void count(String x) {

char[] ch = x.toCharArray();

int letter = 0;

int space = 0;

int num = 0;

int symbol = 0;

int i;

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

if (Character.isLetter(ch[i])) {

letter++;

} else if (Character.isDigit(ch[i])) {

num++;

} else if (Character.isSpaceChar(ch[i])) {

space++;

} else {

symbol++;

}

}

System.out.println("No. of alphabets:" + letter);

System.out.println("No. of digits:" + num);

System.out.println("No. of special symbols:" + symbol);

System.out.println("No. of blank spaces:" + space);

}

public static void word(String str) {

int word = 0;

char ch[] = new char[str.length()];

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

ch[i] = str.charAt(i);

if (((i > 0) && (ch[i] != ' ') && (ch[i - 1] == ' ')) || ((ch[0] != ' ') && (i == 0))) {

word++;

}

}

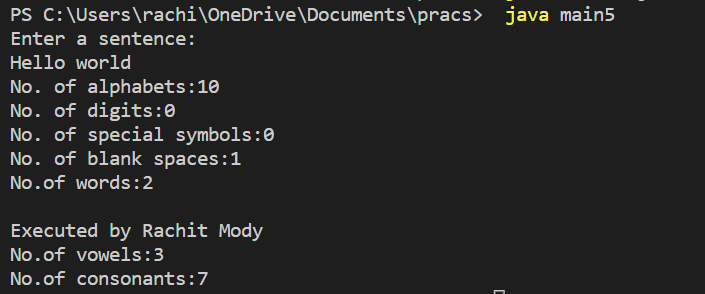
System.out.println("No.of words:" + word);

System.out.println("\nExecuted by Rachit Mody");

}

}

**OUTPUT:**



**CONCLUSION:** Thus we implemented Strings.