1. **PRINT HOW MANY CAPITAL LETTERS,SMALL LETTERS,DIGITS AND SPECIAL CHARACTERS PRESENT IN A STRING**

**package** com.StringPrograms;

**import** java.util.Scanner;

//PRINT HOW MANY CAPITAL LETTERS,SMALL LETTERS,DIGITS AND SPECIAL CHARACTERS PRESENT IN A STRING

**public** **class** DiffTypeCharsSymbols

{

**static** **void** findCharacter(String str) {

**int** uc=0,lc=0,dc=0,spc=0;

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

**char** ch=str.charAt(i);

**if**(ch>='A'&&ch<='Z')

uc++;

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

lc++;

**else** **if**(ch>='0'&&ch<='9')

dc++;

**else**

spc++;

}

System.***out***.println("Count of capital letters "+uc);

System.***out***.println("Count of Small letters "+lc);

System.***out***.println("Count of Digits "+dc);

System.***out***.println("Count of special digits "+spc);

}

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

Scanner sc=**new** Scanner(System.***in***);

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

String str=sc.nextLine();

*findCharacter*(str);

}

}

**2)CALCULATE SUM OF DIGIT IN A STRING**

**package** com.StringPrograms;

//CALCULATE SUM OF DIGIT IN A STRING

**public** **class** StringDigitSum {

**static** **int** digitSum(String str) {

**int** sum=0;

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

**char** ch=str.charAt(i);

**if**(ch>='0'&&ch<='9')

sum=sum+(ch-'0');

}

**return** sum;

}

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

String str="Sandip123Mandal";

**int** ds=*digitSum*(str);

System.***out***.println("Sum of digit in string "+ds);

}

}

**3)COUNT HOW MANY WORDS PRESENT IN A STRING**

**package** com.StringPrograms;

//COUNT HOW MANY WORDS PRESENT IN A STRING

**public** **class** NoOfWords {

**static** **int** countWords(String str) {

**char** ch[]=str.toCharArray();

**int** count=0;

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

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

count++;

}

**return** count;

}

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

String str="I am a very bad boy";

**int** cw=*countWords*(str);

System.***out***.println("Total number of words "+cw);

}

}

**4)COUNT HOW MANY VOWELS AND CONSONANTS PRESENT IN A GIVEN STRING**

**package** com.StringPrograms;

//COUNT HOW MANY VOWELS AND CONSONANTS PRESENT IN A GIVEN STRING

**public** **class** CountVowelsAndConsonants {

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

/\* String str="Java is a Programming Language";

char ch[]=str.toCharArray();

str=str.toLowerCase();

int vc=0;

int cc=0;

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

if(ch[i]>='a'&&ch[i]<='z') {

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

vc++;

}

else

cc++;

}

}

System.out.println("Total Vowels count "+vc);

System.out.println("Total consonants count"+cc);

}

}\*/

String v="aeiouAEIOU";

String str="Apple";

**int** vc=0;

**int** cc=0;

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

**char** ch=str.charAt(i);

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

**if**(v.indexOf(ch)!=-1)

vc++;

**else**

cc++;

}

}

System.***out***.println("Total Vowels count"+vc);

System.***out***.println("Total Consonants count"+cc);

}

}

**5)CONVERT ALL THE CHARACTERS IN A STRING TO UPPERCASE**

**package** com.StringPrograms;

//CONVERT ALL THE CHARACTERS IN A STRING TO UPPERCASE

**public** **class** ConvertUppercase {

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

String str="JAVA IS A HIGH LEVEL LANGUAGE";

**char** ch[]=str.toCharArray();

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

// if(ch[i]>='a'&& ch[i]<='z')

// ch[i]=(char)(ch[i]-32);

// }

// String uc=new String(ch);

// System.out.println(uc);

//CONVERT ALL THE CHARACTER IN A STRING TO LOWERCASE

**if**(ch[i]>='A'&& ch[i]<='Z')

ch[i]=(**char**)(ch[i]+32);

}

String lc=**new** String(ch);

System.***out***.println(lc);

}

}

**6)COVERT IST CHARACTER OF EVERY WORD CAPITAL AND REMAINING ALL ARE SMALL**

**package** com.StringPrograms;

//COVERT IST CHARACTER OF EVERY WORD CAPITAL AND REMAINING ALL ARE SMALL

**public** **class** CovertIstCharacCap {

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

String str="I love jaVa";

**char** []ch=str.toCharArray();

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

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

{

**if**(ch[i]>='a'&&ch[i]<='z')

{

ch[i]=(**char**)(ch[i]-'a'+'A');

}

}

//APART FROM IST CHARACTER IF ANY ONE UPPERCASE

**else** **if**(ch[i]>='A'&&ch[i]<='Z')

{

ch[i]=(**char**)(ch[i]+'a'-'A');

}

}

String uc=**new** String(ch);

System.***out***.println(uc);

}

}

**7)COVERT ALL THE VOWELS TO CAPITALS AND REMAINING ALL ARE SMALL**

**package** com.StringPrograms;

//COVERT ALL THE VOWELS TO CAPITALS AND REMAINING ALL ARE SMALL

**public** **class** ConVowelCapitals {

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

String str="i love java programs";

**char**[]ch=str.toCharArray();

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

{

**if**(ch[i]>='a'&&ch[i]<='z')

{

**if**(ch[i]=='a'||ch[i]=='e'||ch[i]=='i'||ch[i]=='o'||ch[i]=='u')

{

ch[i]=(**char**)(ch[i]-32);

}

}

**else** **if**(ch[i]>='A'&&ch[i]<='Z')

{

**if**(ch[i]!='A'&&ch[i]!='E'&&ch[i]!='I'&&ch[i]!='O'&&ch[i]!='U')

{

ch[i]=(**char**)(ch[i]+32);

}

}

}

String vc=**new** String(ch);

System.***out***.println(vc);

}

}

**8)CONVERT EVERY WORD LAST CHARACTER TO CAPITAL AND REMAINING ALL ARE SMALL**

**package** com.StringPrograms;

//CONVERT EVERY WORD LAST CHARACTER TO CAPITAL AND REMAINING ALL ARE SMALL

**public** **class** ConLastCharacterCap {

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

String str="i love java";

**char** []ch=str.toCharArray();

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

{

**if**(i==ch.length-1&&ch[i]!=' '||ch[i]!=' '&&ch[i+1]==' ')

{

**if**(ch[i]>='a'&&ch[i]<='z')

{

ch[i]=(**char**)(ch[i]-32);

}

}

**else** **if**(ch[i]>='A'&&ch[i]<='Z')

{

ch[i]=(**char**)(ch[i]+32);

}

}

String lc=**new** String(ch);

System.***out***.println(lc);

}

}

**9)Java program to Capitalize the First and Last letter of Each Word of a String**

**package** com.String;

//Java program to Capitalize the First and Last letter of Each Word of a String

**public** **class** CapitalizeFirstAndLastCharacter {

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

String str="i Love Java";

System.***out***.println(*capitalizeCharacter*(str));

}

**private** **static** String capitalizeCharacter(String str) {

**char**[]ch=str.toCharArray();

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

{

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

{

**if**(ch[i]>='a'&&ch[i]<='z')

{

ch[i]=(**char**)(ch[i]-'a'+'A');

}

}

**else** **if**(ch[i]>='A'&ch[i]<='Z')

{

ch[i]=(**char**)(ch[i]+'a'-'A');

}

**if**(i==ch.length-1&&ch[i]!=' '||ch[i]!=' '&&ch[i+1]==' ')

{

**if**(ch[i]>='a'&&ch[i]<='z')

ch[i]=(**char**)(ch[i]-32);

}

}

String rs=**new** String(ch);

**return** rs;

}

}

**9)CONVERT EVERY WORD IST CHARACTER TO ITS LAST CHARACTER**

**package** com.StringPrograms;

//CONVERT EVERY WORD IST CHARACTER TO ITS LAST CHARACTER

**public** **class** SwapIstToLastCha {

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

String str="i love java";

**char**[]ch=str.toCharArray();

**int** f=0;

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

{

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

f=i;

}

**else** **if**(i==ch.length-1&&ch[i]!=' '||ch[i]!=' '&&ch[i+1]==' ')

{

**char** temp=ch[f];

ch[f]=ch[i];

ch[i]=temp;

}

}

String fl=**new** String(ch);

System.***out***.println(fl);

}

}

**10)REVERSE THE GIVEN STRING**

**package** com.StringPrograms;

//REVERSE THE GIVEN STRING

**public** **class** StringReverse {

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

String str="i love java ";

String rs=*reverseString*(str);

System.***out***.println(rs);

}

**static** String reverseString(String str) {

**char**[]ch=str.toCharArray();

**int** f=0;

**int** l=ch.length-1;

**while**(f<l)

{

**char** temp=ch[f];

ch[f]=ch[l];

ch[l]=temp;

f++;

l--;

}

String rs=**new** String(ch);

**return** rs;

}

}

**11)CHECK GIVEN STRING IS PALINDROME OR NOT**

**package** com.StringPrograms;

//CHECK GIVEN STRING IS PALINDROME OR NOT

**public** **class** StringPalindrome {

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

String str="Madam";

String sp=*stringPalindrome*(str);

**if**(str.equals(sp)==**true**)

System.***out***.println("Plindrome string");

**else**

System.***out***.println("Not a palindrome string");

}

**static** String stringPalindrome(String str) {

**char**[]ch=str.toCharArray();

**int** f=0;

**int** l=ch.length-1;

**while**(f<l) {

**char** temp=ch[f];

ch[f]=ch[l];

ch[l]=temp;

f++;

l--;

}

String rs=**new** String(ch);

**return** rs;

}

}

**12)COUNT SPECIFIED CHARACTER IN A GIVEN STRING**

**package** com.StringPrograms;

//COUNT SPECIFIED CHARACTER IN A GIVEN STRING

**public** **class** CountSpecifiedCharacter {

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

String s="sandip";

**int** count=0;

**char** c='a';

//char c[]=s.toCharArray();

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

**if**(s.charAt(i)==c)

count++;

}

System.***out***.println("Total count"+count);

}

}

**13)FREQUENCY OF EACH CHARACTER IN A GIVEN STRING**

**package** com.StringPrograms;

//FREQUENCY OF EACH CHARACTER IN A GIVEN STRING

**public** **class** FrequencyOfCharacter {

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

String str="java@2001";

**int** count[]=**new** **int**[128];

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

**char** ch=str.charAt(i);

count[ch]++;

}

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

**if**(count[i]!=0)

System.***out***.println((**char**)i+"-->"+count[i]);

}

}

}

**14)FREQUENCY OF ONLY ALPHABETS IRRESPECTIVE OF CASE SENSITIVE IN A GIVEN STRING**

**package** com.StringPrograms;

//FREQUENCY OF ONLY ALPHABETS IRRESPECTIVE OF CASE SENSITIVE IN A GIVEN STRING

**public** **class** FrequencyOfAlphabets {

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

String str="AaaBBacc";

**int** count[]=**new** **int**[26];

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

**char** ch=str.charAt(i);

**if**(ch>=65&&ch<=90)

count[ch-65]++;

**else** **if**(ch>=97&&ch<=122)

count[ch-97]++;

}

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

**if**(count[i]!=0)

System.***out***.println((**char**)(i+65)+"-->"+count[i]);

}

}

}

**15)CHECK THE GIVEN STRING IS PANGRAM OR NOT**

**package** com.StringPrograms;

//CHECK THE GIVEN STRING IS PANGRAM OR NOT

//A string is a pangram string if it contains all the character of the alphabets ignoring the case of the alphabets.

/\*Pack my box with five dozen liquor jugs.

The quick brown fox jumps over the lazy dog.

My ex pub quiz crowd gave joyful thanks.

Cozy sphinx waves quart jug of bad milk.

Fix problem quickly with galvanized jets.

\*/

**public** **class** PangramString {

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

String str="the quick brown fox jumps over the lazy dogs";

**boolean** ps=*isPangram*(str);

**if**(ps)

System.***out***.println("String is pangram");

**else**

System.***out***.println("String is not pangram");

}

**private** **static** **boolean** isPangram(String str) {

**if**(str.length()<26)

**return** **false**;

**int** ct[]=*countFreqAlphabet*(str);

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

**if**(ct[i]==0)

**return** **false**;

}

**return** **true**;

}

**private** **static** **int**[] countFreqAlphabet(String str) {

**int** count[]=**new** **int**[26];

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

**char** ch=str.charAt(i);

**if**(ch>=65&&ch<=90)

count[ch-65]++;

**else** **if**(ch>=90&&ch<=122)

count[ch-97]++;

}

**return** count;

}

}

**16)CHECK THE GIVEN STIRNG IS PANGRAM OR NOT USING PREDEFINED METHOD**

**package** com.StringPrograms;

//CHECK THE GIVEN STIRNG IS PANGRAM OR NOT USING PREDEFINED METHOD

**public** **class** StringPangram {

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

String str="the quick brown fox jumps over the lazy dogs";

**boolean** ps=*isPangram*(str);

**if**(ps)

System.***out***.println("String is pangram");

**else**

System.***out***.println("String is not pangram");

}

**private** **static** **boolean** isPangram(String str) {

str=str.toLowerCase();

**for**(**char** ch='a';ch<='z';ch++) {

**if**(str.indexOf(ch)==-1)

**return** **false**;

}

**return** **true**;

}

**private** **static** **int**[] countFreqAlphabet(String str) {

**int** count[]=**new** **int**[26];

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

**char** ch=str.charAt(i);

**if**(ch>=65&&ch<=90)

count[ch-65]++;

**else** **if**(ch>=90&&ch<=122)

count[ch-97]++;

}

**return** count;

}

}

**17)CHECK WHEATHER TWO STRING ARE ANAGRAM OR NOT**

**package** com.StringPrograms;

//CHECK WHEATHER TWO STRING ARE ANAGRAM OR NOT

//Two strings are called anagrams if they contain same set of characters but in different order.

//"keep ? peek", "Mother In Law - Hitler Woman".

**public** **class** AnagramString {

**private** **static** **int**[] countFreqAlphabet(String str) {

**int** count[]=**new** **int**[26];

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

**char** ch=str.charAt(i);

**if**(ch>=65&&ch<=90)

count[ch-65]++;

**else** **if**(ch>=90&&ch<=122)

count[ch-97]++;

}

**return** count;

}

**private** **static** **boolean** isAnagram(String str1,String str2) {

**int** []c1=*countFreqAlphabet*(str1);

**int** []c2=*countFreqAlphabet*(str2);

**for**(**int** i=0;i<26;i++) {

**if**(c1[i]!=c2[i])

**return** **false**;

}

**return** **true**;

}

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

String str1="Mother in law";

String str2="Hitler Woman";

**boolean** as=*isAnagram*(str1,str2);

**if**(as)

System.***out***.println("Strings are anagram");

**else**

System.***out***.println("Strings are not anagram");

}

}

**18)CHECK TWO STRING ARE ANAGRAM OR NOT BY USING PREDEFINED METHOD**

**package** com.StringPrograms;

**import** java.util.Arrays;

//CHECK TWO STRING ARE ANAGRAM OR NOT BY USING PREDEFINED METHOD

**public** **class** StringAnagram {

**private** **static** **int**[] countFreqAlphabet(String str) {

**int** count[]=**new** **int**[26];

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

**char** ch=str.charAt(i);

**if**(ch>=65&&ch<=90)

count[ch-65]++;

**else** **if**(ch>=90&&ch<=122)

count[ch-97]++;

}

**return** count;

}

**private** **static** **boolean** isAnagram(String str1,String str2) {

str1=str1.replaceAll(" ","");

str2=str2.replaceAll(" ", "");

**if**(str1.length()!=str2.length())

**return** **false**;

str1=str1.toLowerCase();

str2=str2.toLowerCase();

**char**[]c1=str1.toCharArray();

**char**[]c2=str2.toCharArray();

Arrays.*sort*(c1);

Arrays.*sort*(c2);

**return** Arrays.*equals*(c1,c2);

}

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

String str1="Sandip";

String str2="Mandal";

**boolean** as=*isAnagram*(str1,str2);

**if**(as)

System.***out***.println("Strings are anagram");

**else**

System.***out***.println("Strings are not anagram");

}

}

**19)Java Program to find length of the string without using length function**

**package** com.String;

//Java Program to find length of the string without using length function

**public** **class** FindStringLength {

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

String str="Sandip Mandal";

System.***out***.println(*stringLength*(str));

}

**private** **static** **int** stringLength(String str) {

**int** length=0;

**for**(**char** c:str.toCharArray())

length++;

**return** length;

}

}

**20)Java program to toggle each character in a string**

**package** com.String;

//Java program to toggle each character in a string

//If the letter is in uppercase we will convert it to lowercase and

//if it is in uppercase we will convert it to lowercase.

**public** **class** StringToggle {

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

String str="i lOvE JavA";

String s=*stringToggle*(str);

System.***out***.println("Toggle String is \n"+s);

}

**private** **static** String stringToggle(String str) {

**char** ch[]=str.toCharArray();

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

{

**if**(ch[i]>='a'&&ch[i]<='z')

ch[i]=(**char**)(ch[i]-32);

**else** **if**(ch[i]>='A'&&ch[i]<='Z')

ch[i]=(**char**)(ch[i]+32);

}

String rs=**new** String(ch);

**return** rs;

}

}

**21)REMOVE VOWELS FROM A GIVEN STRING**

**package** com.String;

**import** java.util.\*;

**public** **class** RemoveVowels {

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

Scanner sc =**new** Scanner(System.***in***);

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

String s = sc.nextLine();

String str="";

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

{

**char** ch=s.charAt(i);

**if**(ch!='a' && ch!='e' && ch!='i' && ch!='o' &&

ch!='u' &&ch!='A' && ch!='E' && ch!='I' &&

ch!='O' && ch!='U')

str=str+ch;

}

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

}

}

**22)Java Program to Remove Vowels from a String**

**package** com.String;

//Java Program to Remove Vowels from a String

**public** **class** VowelRemove {

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

String s="I Love Java Programming";

String vr=*vowelRemove*(s);

System.***out***.println("String After Removing vowels\n "+vr);

}

**private** **static** String vowelRemove(String s) {

String str=" ";

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

{

**char** ch=s.charAt(i);

**if**(ch!='a' && ch!='e' && ch!='i' && ch!='o' &&

ch!='u' &&ch!='A' && ch!='E' && ch!='I' &&

ch!='O' && ch!='U')

str=str+ch;

}

String rs=**new** String(str);

**return** rs;

}

}

**23)Java Program to Remove Consonants from a String**

**package** com.String;

//Java Program to Remove Consonants from a String

**public** **class** RemoveConsonants {

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

String s="I Love Java Programming";

String vr=*consonantsRemove*(s);

System.***out***.println("String After Removing consonants\n "+vr);

}

**private** **static** String consonantsRemove(String s) {

String str=" ";

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

{

**char** ch=s.charAt(i);

**if**(ch=='a' || ch=='e' || ch=='i' || ch=='o' ||

ch=='u' ||ch=='A'||ch=='E' || ch=='I' ||

ch=='O' || ch=='U')

str=str+ch;

}

String rs=**new** String(str);

**return** rs;

}

}

**24)Java program to Remove spaces from a string**

**package** com.String;

//Java program to Remove spaces from a string

**public** **class** RemoveSpace {

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

String str="I Love Java";

String rs=*removeSpace*(str);

System.***out***.println("String after remove spaces \n"+rs);

}

**private** **static** String removeSpace(String str) {

String s="";

**char**[]ch=str.toCharArray();

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

{

**if**(ch[i]!=' ')

s=s+ch[i];

}

**return** **new** String(s);

}

}

**package** com.String;

//Java program to Remove spaces from a string

**public** **class** RemoveSpace {

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

String str="I Love Java";

String rs=*removeSpace*(str);

System.***out***.println("String after remove spaces \n"+rs);

}

**private** **static** String removeSpace(String str) {

String s="";

**char**[]ch=str.toCharArray();

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

{

**if**(ch[i]!=' ')

s=s+ch[i];

}

**return** **new** String(s);

}

}

**25)Java program to Find All Non-Repeating character in a String**

**package** com.String;

//Java program to Find All Non Repeating character in a String

**public** **class** FindNonRepeatingCharacters {

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

String str="JAVA LANGUAGE";

System.***out***.println("Non Repeating characters are");

*nonRepeatingCharacter*(str);

}

**private** **static** **void** nonRepeatingCharacter(String str) {

**int** count[]=**new** **int**[128];

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

{

**char** c=str.charAt(i);

count[c]++;

}

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

{

**if**(count[i]==1) {

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

}

}

}

}

**26)Java program to Find First non repeating character in a String**

**package** com.String;

//Java program to Find First non repeating character in a String

**public** **class** FindFirstNonRepeatingCharacter {

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

String str="ABBAACCDCA";

*nonRepeatingCharacter*(str);

}

**private** **static** **void** nonRepeatingCharacter(String str) {

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

{

**boolean** unique=**true**;

**for**(**int** j=0;j<str.length();j++)

{

**if**(i!=j&&str.charAt(i)==str.charAt(j))

{

unique=**false**;

}

}

**if**(unique)

{

System.***out***.println(str.charAt(i));

**break**;

}

}

}

}

**27)FIND FREQUENCY OF ONLY EACH ALPHABETS OF A STRING IRRESPECTIVE CASE SENSETIVE**

**package** com.String;

//FIND FREQUENCY OF ONLY EACH ALPHABETS OF A STRING IRRESPECTIVE CASE SENSETIVE

**public** **class** FrequencyOfEachAlphabets {

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

String str="AbbACC";

*countFrequency*(str);

}

**private** **static** **void** countFrequency(String str) {

**int** count[]=**new** **int**[26];

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

{

**char** ch=str.charAt(i);

**if**(ch>=65&&ch<=90)

count[ch-65]++;

**else** **if**(ch>=97&&ch<=122)

count[ch-97]++;

}

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

{

**if**(count[i]!=0)

System.***out***.println((**char**)(i+65)+"-->"+count[i]);

}

}

}

**28)Java program to find the duplicate characters in a string**

**package** com.String;

//Java program to find the duplicate characters in a string

**public** **class** DuplicateCharactersInString {

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

String str="ABBCCDEF";

System.***out***.println("Duplicates characters in string");

*findDuplicates*(str);

}

**private** **static** **void** findDuplicates(String str) {

**int** count[]=**new** **int**[128];

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

{

**char** ch=str.charAt(i);

count[ch]++;

}

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

{

**if**(count[i]>1)

System.***out***.println((**char**)i+" ");

}

}

}

**29)Java program to remove the duplicate characters in a string**

**package** com.String;

//Java program to remove the duplicate characters in a string

**public** **class** RemoveDuplicatesInString {

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

String str="AABBCCDDEE";

String dc=*removeDuplicates*(str);

System.***out***.println("After removing duplicates characters string becomes\n "+dc);

}

**private** **static** String removeDuplicates(String str) {

String newString="";

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

{

**char** ch=str.charAt(i);

**if**(newString.indexOf(ch)==-1)

newString=newString+ch;

}

**return** **new** String(newString);

}

}

**30)SWAPPING TWO STRING WITHOUT USING THIRD VARIABLE**

**package** com.String;

//SWAPPING TWO STRING WITHOUT USING THIRD VARIABLE

**public** **class** SwapTwoString {

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

String a="hello";

String b="world";

System.***out***.println("String before swapping:\n a="+a+" b="+b);

*swappingTwoString*(a,b);

}

**private** **static** **void** swappingTwoString(String a, String b) {

a=a+b;

b=a.substring(0,a.length()-b.length());

a=a.substring(b.length());

System.***out***.println("String after swapping:\n a="+a+" b="+b);

}

}

**package** com.String;

//SWAPPING TWO STRING WITHOUT USING THIRD VARIABLE

**public** **class** StringSwapping {

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

String a="Hello";

String b="World";

System.***out***.println("String before Swapping:\n a="+a+" b="+b);

*swappingString*(a,b);

}

**private** **static** **void** swappingString(String a, String b) {

**char**[] c1=a.toCharArray();

**char**[] c2=b.toCharArray();

**for**(**int** i=0;i<c1.length&&i<c2.length;i++)

{

c1[i]^=c2[i];//c1[i]=(char) (c1[i]^c2[i]);

c2[i]^=c1[i];//c2[i]=(char) (c2[i]^c1[i]);

c1[i]^=c2[i];//c1[i]=(char) (c1[i]^c2[i]);

}

a=**new** String(c1);

b=**new** String(c2);

System.***out***.println("String after swapping:\n a="+a+" b "+b);

}

}

**31)REVERSE THE SENTENCE BY WORDS IN JAVA**

**package** com.String;

//REVERSE THE SENTENCE BY WORDS IN JAVA

//I LOVE JAVA PROGRAMMING--->PROGRAMMING JAVA LOVE I

**public** **class** ReverseSentence {

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

String str="I Love Java Programming";

String rs=*reverseSentence*(str);

System.***out***.println("Reverse sentence is: "+rs);

}

**private** **static** String reverseSentence(String str) {

str=str+" ";

String word="";

String rev="";

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

{

**char** ch=str.charAt(i);

**if**(ch!=' ')

{

word=word+ch;

}

**else**

{

rev=word+" "+rev;

word=" ";

}

}

**return** **new** String(rev);

}

}

**32)REVERSE EACH WORD IN A SENTENCE**

**package** com.String;

//REVERSE EACH WORD IN A SENTENCE

//I LOVE JAVA-->I EVOL AVAJ

**public** **class** ReverseEachWordInSentence {

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

String str="I Love Java Programming";

String rw=*reverseEachWords*(str);

System.***out***.println("After reverse each word in a sentence \n"+rw);

}

**private** **static** String reverseEachWords(String str) {

String word="";

String rev="";

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

{

**char** ch=str.charAt(i);

**if**(ch==' ')

{

rev=rev+*isReverse*(word)+" ";

word="";

**continue**;

}

word=word+ch;

}

rev=rev+*isReverse*(word);

**return** **new** String(rev);

}

**private** **static** String isReverse(String str) {

**char** ch[]=str.toCharArray();

**int** f=0;

**int** l=str.length()-1;

**while**(f<l)

{

**char** temp=ch[f];

ch[f]=ch[l];

ch[l]=temp;

f++;

l--;

}

**return** **new** String(ch);

}

}

**33)CHECK A STRING IS SUBSTRING OF ANOTHER STRING OR NOT**

**package** com.String;

//CHECK A STRING IS SUBSTRING OF ANOTHER STRING OR NOT

**public** **class** FindSubstring {

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

String s1="I Love Java";

String s2="Love";

**boolean** rs=*findSubstring*(s1,s2);

**if**(rs)

System.***out***.println("Substring is present");

**else**

System.***out***.println("Substring is not present");

}

**private** **static** **boolean** findSubstring(String s1, String s2) {

**if**(s1.indexOf(s2)!=-1)

**return** **true**;

**else**

**return** **false**;

}

}

**34)PRINT ALL SUBSTRING OF A GIVEN STRING**

**package** com.String;

//PRINT ALL SUBSTRING OF A GIVEN STRING

**public** **class** PrintAllSubStrings {

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

String str="abcd";

*findSubstring*(str);

}

**private** **static** **void** findSubstring(String str) {

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

{

**for**(**int** j=i;j<str.length();j++)

{

System.***out***.print(str.substring(i,j+1)+" ");

}

}

}

}

**35)Java Program to Count the Total Number of Punctuation Characters Exists in a String**

**package** com.StringPrograms;

//Java Program to Count the Total Number of Punctuation Characters Exists in a String

**public** **class** CountPunctuation {

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

String str="He said, 'The mailman loves you'. I heard it with my own ears.";

**int** cp=*countPunctuation*(str);

System.***out***.println("The number of punctuations exists in the string is:"+cp);

}

**private** **static** **int** countPunctuation(String str) {

**int** count=0;

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

{

**if**(str.charAt(i) == '!' || str.charAt(i) == ',' || str.charAt(i) == ';' || str.charAt(i) == '.' || str.charAt(i) == '?' || str.charAt(i) == '-' ||

str.charAt(i) == '\'' || str.charAt(i) == '\"' || str.charAt(i) == ':')

count++;

}

**return** count;

}

}

**36)Java Program to divide a string in 'N' equal parts**

**package** com.StringPrograms;

//Java Program to divide a string in 'N' equal parts

**public** **class** StringDivide {

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

String str="aabbccc";

*divideString*(str);

}

**private** **static** **void** divideString(String str) {

//n determines the variable that divide the string in 'n' equal parts

**int** n=3,k=0;

**int** chars=str.length()/n;

String eqlStr[]=**new** String[n];

**if**(str.length()%n!=0)

System.***out***.println("String cannot be divide into "+n+" parts");

**else**

{

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

{

//Dividing string in n equal part using substring()

String part=str.substring(i,i+chars);

eqlStr[k]=part;

k++;

}

System.***out***.println(n+"Equals parts of given string are");

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

{

System.***out***.print(eqlStr[i]+" ");

}

System.***out***.println();

}

}

}

**37)Java Program to replace the spaces of a string with a specific character**

**package** com.StringPrograms;

//Java Program to replace the spaces of a string with a specific character

**public** **class** ReplaceStringWithSpecificChar {

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

String str="I Love Java Programming";

String rs=*replaceString*(str);

System.***out***.println("String after replaceing spaces \n"+rs);

}

**private** **static** String replaceString(String str) {

String s="";

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

{

**char** ch=str.charAt(i);

**if**(ch==' ')

s=s+'-';

**else**

s=s+ch;

}

**return** **new** String(s);

}

}

**38)Java Program to determine whether one string is a rotation of another**

**package** com.StringPrograms;

//Java Program to determine whether one string is a rotation of another

**public** **class** CheckStringRotation {

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

String str1="abcde";

String str2="abced";

System.***out***.println(*isRotation*(str1,str2));

}

**private** **static** **boolean** isRotation(String str1, String str2) {

String str3=str1+str1;

**if**(str1.length()!=str2.length())

{

**return** **false**;

}

**if**(str3.indexOf(str2)==-1)

{

**return** **false**;

}

**else** {

**return** **true**;

}

}

}

**39) Find maximum occurring character in a string**

**package** com.StringPrograms;

//Find maximum occurring character in a string

**public** **class** MaximumOccurringCharacter {

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

String str="abbbccda";

**char** mi=*minOccurringCharacter*(str);

System.***out***.println("Minimum occuring character "+mi);

}

**private** **static** **char** minOccurringCharacter(String str) {

**int** count[]=**new** **int**[128];

**int** max=-1;

**char** result=' ';

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

{

count[str.charAt(i)]++;

}

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

{

**if**(max<count[str.charAt(i)])

{

max=count[str.charAt(i)];

result=str.charAt(i);

}

}

**return** result;

}

}

**40)Java Program to find maximum and minimum occurring character in a string.**

**package** com.StringPrograms;

//Java Program to find maximum and minimum occurring character in a string.

**public** **class** MaxAndMinOccurringCharacters {

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

String str="abcaaadce";

*findMaxAndMinCharacter*(str);

}

**private** **static** **void** findMaxAndMinCharacter(String str) {

**int** freq[]=**new** **int**[str.length()];

**char** charArray[]=str.toCharArray();

**int** i,j;

**int** max,min;

**char** maxChar=str.charAt(0);

**char** minChar=str.charAt(0);

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

{

freq[i]=1;

**for**(j=i+1;j<str.length();j++)

{

**if**(charArray[i]==charArray[j]&&charArray[i]!=' '&&charArray[i]!='0')

{

freq[i]++;

freq[j]='0';

}

}

}

min=max=freq[0];

**for**( i=0;i<freq.length;i++)

{

**if**(min>freq[i]&&freq[i]!='0')

{

min=freq[i];

minChar=charArray[i];

}

**else** **if**(max<freq[i])

{

max=freq[i];

maxChar=charArray[i];

}

}

System.***out***.println("Maximum character "+maxChar);

System.***out***.println("Minimum character "+minChar);

}

}

**41)Java program to find the duplicate words in a string**

**package** com.StringPrograms;

//Java program to find the duplicate words in a string

**public** **class** FindDuplicateWords {

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

String str="I love java programming language java";

System.***out***.println("Duplicates words are");

*findDuplicates*(str);

}

**private** **static** **void** findDuplicates(String str) {

str=str.toLowerCase();

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

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

{

**int** count=1;

**for**(**int** j=i+1;j<words.length;j++)

{

**if**(words[i].equals(words[j]))

{

count++;

//Set words[j] to 0 to avoid printing visited word

words[j]="0";

}

}

**if**(count>1&&words[i]!="0")

System.***out***.println(words[i]+" ");

}

}

}

**42)FIND TOTAL NUMBER OF DUPLICATES WORDS PRESENT IN A GIVEN STRING**

**package** com.StringPrograms;

//FIND TOTAL NUMBER OF DUPLICATES WORDS PRESENT IN A GIVEN STRING

**public** **class** CountDuplicatesWords {

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

String str="I love java programming java language java";

*countDuplicates*(str);

//System.out.println("Total number of duplicates words are "+c);

}

**private** **static** **void** countDuplicates(String str) {

str=str.toLowerCase();

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

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

{

**int** count=1;

**for**(**int** j=i+1;j<words.length;j++)

{

**if**(words[i].equals(words[j]))

{

count++;

words[j]="0";

}

}

**if**(count>1&&words[i]!="0")

System.***out***.println("Total number of duplicates words are "+count);

}

}

}

**43)Java Program to find the largest and smallest word in a string**.

**package** com.StringPrograms;

//Java Program to find the largest and smallest word in a string.

**public** **class** FindLargestAndSmallestWord {

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

String str="I love java language";

*findLargestSmallest*(str);

}

**private** **static** **void** findLargestSmallest(String str) {

str=str+" ";

String word="",longword="",smallword="";

**int** longlen=0,wordlen=0;

**int** smalllen=str.length();

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

{

**char** ch=str.charAt(i);

**if**(ch!=' ')

{

word=word+ch;

}

**else**

{

wordlen=word.length();

**if**(wordlen>longlen)

{

longlen=wordlen;

longword=word;

}

**if**(wordlen<smalllen)

{

smalllen=wordlen;

smallword=word;

}

word=" ";

}

}

System.***out***.println("Longest word is "+longword+" and length is "+longlen);

System.***out***.println("Smallest word is "+smallword+" and length is "+smalllen);

}

}

**44)CHECK THE GIVEN STRING WORD IS FUNNY OR NOT**

**package** com.StringPrograms;

//CHECK THE GIVEN STRING WORD IS FUNNY OR NOT

**public** **class** CheckFunnyWords {

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

String str="bdwy";

**boolean** rs=*isFunny*(str);

**if**(rs)

System.***out***.println("String word is funny");

**else**

System.***out***.println("String is not funny");

}

**private** **static** **boolean** isFunny(String str) {

**int** i=1;

**int** j=str.length()-2;

str=str.toLowerCase();

**while**(i<=j)

{

**if**((Math.*abs*(str.charAt(i)-str.charAt(i-1)))!=Math.*abs*(str.charAt(j)-(str.charAt(j+1))))

**return** **false**;

i++;

j--;

}

**return** **true**;

}

}

**45)Print all funny words in a string**

**package** com.StringPrograms;

//Print all funny words in a string

/\*What is a funny word ?

Reverse the given string. Iterate through each character of that string,

compare the absolute difference in the ASCII values of the characters at positions

0 and 1, 1 and 2, 2 and 3 and so on to the end.

If the list of absolute differences is the same for both strings,

they are funny otherwise not.

\*/

**public** **class** FindFunnyWordsInString {

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

String str="Miss Arora teaches us malayalam bdwy";

*printFunnyWords*(str);

}

**private** **static** **void** printFunnyWords(String str) {

str = str + " ";

String word = "";

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

**char** ch = str.charAt(i);

**if** (ch != ' ')

word = word + ch;

**else** {

**if** (*checkFunny*(word))

System.***out***.println(word);

word = "";

}

}

}

**private** **static** **boolean** checkFunny(String word) {

**int** i = 1;

**int** j = word.length() - 2;

word = word.toLowerCase();

**while** (i <= j) {

**if** ((Math.*abs*(word.charAt(i) - word.charAt(i - 1))) !=

Math.*abs*((word.charAt(j) - word.charAt(j + 1))))

**return** **false**;

i++;

j--;

}

**return** **true**;

}

}

**46)Java program to print Even length words in a String**

**package** com.StringPrograms;

//Java program to print Even length words in a String

/\*Input: s = "This is a java language"

Output: This

is

java

language

\*/

**public** **class** PrintEvenLengthWords {

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

String str="I love java";

*evenLengthWords*(str);

}

**private** **static** **void** evenLengthWords(String str) {

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

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

{

**if**(words[i].length()%2==0)

System.***out***.println(words[i]+" ");

}

}

}

**47)Insert a String into another String in Java**

**package** com.StringPrograms;

//Insert a String into another String in Java

**public** **class** InsertNewString {

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

String str="Greeksgreeks";

String is="for";

**int** index=5;

String s=*insertString*(str,is,index);

System.***out***.println("After inserting string becomes "+s);

}

**private** **static** String insertString(String str, String is, **int** index) {

String s=" ";

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

{

s=s+str.charAt(i);

**if**(i==index)

s=s+is;

}

**return** **new** String(s);

}

}

**48)Swapping Pairs of Characters in a String in Java**

**package** com.StringPrograms;

//Swapping Pairs of Characters in a String in Java

/\*The given string contains even number of characters.

\* Therefore, we swap every pair of characters.

\* The given string contains odd number of characters.

\* Therefore, we swap every pair of characters and last character remains as it is.

\*/

**public** **class** SwappingStringCharacterPairs {

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

String str="GeeksForGeeks";

String sp=*swappingPairs*(str);

System.***out***.println("After swapping chracters string becomes "+sp);

}

**private** **static** String swappingPairs(String str){))

            return str;

**if** (str == **null** || str.isEmpty())

**return** str;

**char** []ch=str.toCharArray();

**for**(**int** i=0;i<ch.length-1;i+=2)

{

**char** temp=ch[i];

ch[i]=ch[i+1];

ch[i+1]=temp;

}

**return** **new** String(ch);

}

}

**49)Replace a character at a specific index in a String in Java**

**package** com.StringPrograms;

//Replace a character at a specific index in a String in Java

/\*Input: String = "Geeks Gor Geeks", index = 6, ch = 'F'

Output: "Geeks For Geeks."

\*/

**public** **class** ReplaceStringCharacter {

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

String str="Geeks gor Geeks";

**char** ch='F';

**int** index=6;

String rs=*replaceCharacter*(str,ch,index);

System.***out***.println("Replaceing character string becomes "+rs);

}

**private** **static** String replaceCharacter(String str,**char** ch,**int** index) {

str=str.substring(0,index)+ch+str.substring(index+1);

**return** **new** String(str);

}

}

**50)Remove Zeros From String in Java**

**package** com.StringPrograms;

//Remove Zeros From String in Java

**public** **class** RemoveZeroFromZeros {

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

String str="1002035021";

System.***out***.println(*removeZero*(str));

}

**private** **static** String removeZero(String str) {

String s="";

**if**(str.length()==0||str.isEmpty())

**return** **null**;

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

{

**char** ch=str.charAt(i);

**if**(ch!='0')

s=s+ch;

}

**return** **new** String(s);

}

}

**51)Remove Leading Zeros From String in Java**

**package** com.StringPrograms;

//Remove Leading Zeros From String in Java

**public** **class** RemoveLeadingZeros {

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

String str="000012314";

System.***out***.println(*removeLeadingZeros*(str));

}

**private** **static** String removeLeadingZeros(String str) {

String s="";

**int** index=0;

**if**(str.length()==0||str.isEmpty())

**return** **null**;

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

{

**char** ch=str.charAt(i);

**if**(ch!='0') {

index=i;

**break**;

}

}

s=str.substring(index,str.length());

**return** **new** String(s);

}

}

**52)REMOVE TRAILING ZEROS FROM A GIVEN STRING**

**package** com.StringPrograms;

//REMOVE TRAILING ZEROS FROM A GIVEN STRING

**public** **class** RemoveTrailingZeros {

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

String str="12345600000";

System.***out***.println(*removeTrailingZeros*(str));

}

**private** **static** String removeTrailingZeros(String str) {

**if**(str.length()==0||str.isEmpty())

**return** **null**;

String s="";

**int** index=str.length()-1;

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

{

**char** ch=str.charAt(i);

**if**(ch!='0')

{

index=i;

**break**;

}

}

**return** s=str.substring(0,index+1);

}

}

**53)JAVA PROGRAM TO SORT THE GIVEN STRING**

**package** com.StringPrograms;

//JAVA PROGRAM TO SORT THE GIVEN STRING

**public** **class** StringSorting {

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

String str="GeeksForGeeks";

String ss=*sortString*(str);

System.***out***.println("After sorting string becomes "+ss);

}

**private** **static** String sortString(String str) {

**if**(str.length()==0||str.isEmpty())

**return** **null**;

str=str.toLowerCase();

**char**[]ch=str.toCharArray();

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

{

**for**(**int** j=i+1;j<ch.length;j++)

{

**if**(ch[i]>ch[j]) {

**char** temp=ch[i];

ch[i]=ch[j];

ch[j]=temp;

}

}

}

**return** **new** String(ch);

}

}

**54)Print first and last letter of each word in a string**

**package** com.StringPrograms;

//Print first and last letter of each word in a string

**public** **class** PrintFirstLetterEachWord {

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

String str="Geeks for geeks";

*printFirstLetter*(str);

}

**private** **static** **void** printFirstLetter(String str) {

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

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

{

String s=words[i];

System.***out***.println(s.charAt(0)+" "+s.charAt(s.length()-1));

}

}

}

**55)Java program to Compare two strings lexicographically**

**package** com.StringPrograms;

//Java program to Compare two strings lexicographically

/\*Using user-defined function : Define a function to compare values with following conditions :

if (string1 > string2) it returns a positive value.

if both the strings are equal lexicographically

i.e.(string1 == string2) it returns 0.

if (string1 < string2) it returns a negative value.

The value is calculated as (int)str1.charAt(i) – (int)str2.charAt(i)\*/

**public** **class** CompareToStrings {

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

String str1="GeeksForGeeks";

String str2="Geeks";

System.***out***.println(*compareString*(str1,str2));

}

**private** **static** **int** compareString(String str1, String str2) {

**int** l1=str1.length();

**int** l2=str2.length();

**int** lmin=Math.*min*(l1, l2);

**for**(**int** i=0;i<lmin;i++)

{

**int** str1\_char=(**int**)str1.charAt(i);

**int** str2\_char=(**int**)str2.charAt(i);

**if**(str1\_char!=str2\_char) {

**return** str1\_char - str2\_char;

}

}

**if**(l1!=l2)

{

**return** l1-l2;

}

//If none of the above conditions is true,

// it implies both the strings are equal

**else**

{

**return** 0;

}

}

}

**56)JAVA PROGRAM TO SORT ELEMENTS IN LEXICOGRAPHICAL ORDER(DICTIONARY ORDER)**

**package** com.StringPrograms;

//JAVA PROGRAM TO SORT ELEMENTS IN LEXICOGRAPHICAL ORDER(DICTIONARY ORDER)

**public** **class** LexicographicallyStringComparison {

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

String words[]= {"Sandip","Manik","Ramchandra"};

*stringSorting*(words);

}

**private** **static** **void** stringSorting(String[] words) {

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

{

**for**(**int** j=i+1;j<words.length;j++)

{

**if**(words[i].compareTo(words[j])>0)

{

String temp=words[i];

words[i]=words[j];

words[j]=temp;

}

}

}

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

{

System.***out***.println(words[i]+" ");

}

}

}

**57)Replace the substring with another string without using built-in method in Java**

**package** com.StringPrograms;

//Replace the substring with another string without using built-in method in Java

**public** **class** ReplaceingSubString {

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

String str="I love java";

String oldWord="java";

String newWord="java programming";

String rs=*repelaceingSubString*(str,oldWord,newWord);

System.***out***.println("After replaceing modified String becomes "+rs);

}

**private** **static** String repelaceingSubString(String str, String oldWord, String newWord) {

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

String [] temp=**new** String[words.length];

String result="";

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

{

**if**(words[i].equals(oldWord))

temp[i]=newWord;

**else**

temp[i]=words[i];

}

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

{

result=result+temp[i]+" ";

}

**return** **new** String(result);

}

}

**58)Program in Java to Replace a Particular Word From a Given String**

**package** com.StringPrograms;

//Program in Java to Replace a Particular Word From a Given String

**public** **class** RemoveWordFromString {

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

String str="I love java";

String oldWord="java";

String rs=*removeWord*(str,oldWord);

System.***out***.println("After removing word String becomes "+rs);

}

**private** **static** String removeWord(String str, String oldWord) {

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

String s="";

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

{

**if**(!words[i].equals(oldWord))

s=s+words[i]+" ";

}

**return** **new** String(s);

}

}