**Assignment-8**

**By: Kaviya.C**

**1. Playing with String - I**

**Given a string array and non negative integer (n) apply the following rules.**

**1. Pick nth character from each String element in the String array and form a new String.**

**2. If nth character not available in a particular String in the array**

**consider $ as the character.**

**3. Return the newly formed string.**

**Include a class UserMainCode with a static method formString which accepts the string and integer. The return type is the string formed based on rules.**

**Create a Class Main which would be used to accept the string and integer and call the static method present in UserMainCode.**

**Input and Output Format:**

**Input consists of a an integer which denotes the size of the array**

**followed by the array of strings and an integer (n).**

**Output consists of a string .**

**Refer sample output for formatting specifications.**

**Sample Input 1:**

**4**

**ABC**

**XYZ**

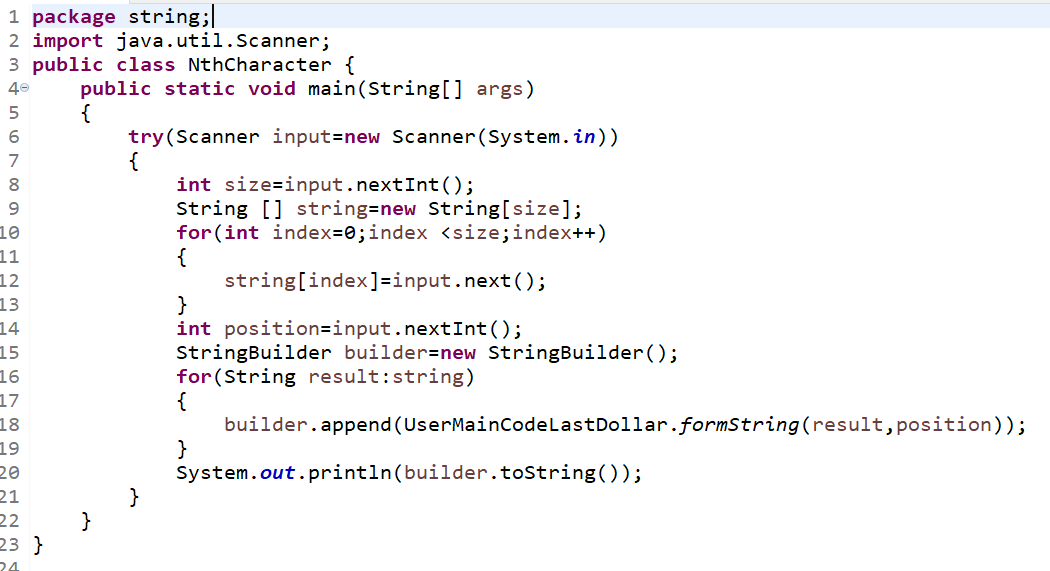
**EFG**

**MN**

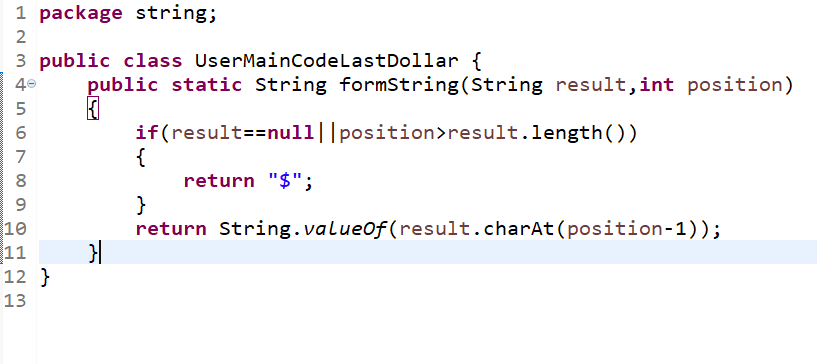
**3**

**Sample Output 1:**

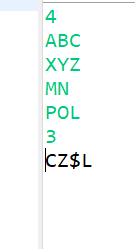
**CZG$**

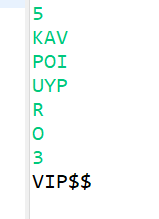
****

**UserMainCode:**

****

**Output:**

****

****

**2. Reverse SubString**

**Given a string, startIndex and length, write a program to extract the**

**substring from right to left. Assume the last character has index 0.**

**Include a class UserMainCode with a static method “reverseSubstring” that**

**accepts 3 arguments and returns a string. The 1st argument corresponds to**

**the string, the second argument corresponds to the startIndex and the**

**third argument corresponds to the length.**

**Create a class Main which would get a String and 2 integers as input and**

**call the static method reverseSubstring present in the UserMainCode.**

**Input and Output Format:**

**The first line of the input consists of a string.**

**The second line of the input consists of an integer that corresponds to**

**the startIndex.**

**The third line of the input consists of an integer that corresponds to**

**the length of the substring.**

**Sample Input:**

**rajasthan**

**2**

**3**

**Sample Output:**

**Hts**

**Ans:**

**package** string;

**import** java.util.Scanner;

**public** **class** ReverseString {

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

{

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

System.***out***.println("Enter the word: ");//rameshwaram

String input1=input.nextLine();

System.***out***.println("enter the integer: ");//2

**int** number=input.nextInt();

System.***out***.println("enter the how many letter want: ");//3

**int** letterCount=input.nextInt();

System.***out***.println(*reverseString*(input1,number,letterCount));

input.close();

}

**public** **static** String reverseString(String input1,**int** number,**int** letterCount)

{

StringBuffer string=**new** StringBuffer(input1);//rajasthan//rameshwaram

string.reverse();//nahtsajar//matawhsemar

String output=string.substring(number,number+letterCount);//2---h from this need 3 letter hts//2----taw

**return** output;

}

}

Enter the word:

rajasthan

enter the integer:

2

enter the how many letter want:

3

hts

**3. Fetching Middle Characters from String**

**Write a program to read a string of even length and to fetch two middle**

**most characters from the input string and return it as string output.**

**Include a class UserMainCode with a static method getMiddleChars which**

**accepts a string of even length as input . The return type is a string**

**which should be the middle characters of the string.**

**Create a class Main which would get the input as a string and call the**

**static method getMiddleChars present in the UserMainCode.**

**Input and Output Format:**

**Input consists of a string of even length.**

**Output is a string .**

**Refer sample output for formatting specifications.**

**Sample Input 1:**

**this**

**Sample Output 1:**

**Hi**

**Ans:**

**package** string;

**import** java.util.Scanner;

**public** **class** FetchMiddleCharacter {

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

{

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

System.***out***.println("enter the word having even digits ended...like helloo etc! : ");

String word=input.nextLine();//this

String result=UserMainCode.*getMiddleCharacter*(word);//this

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

input.close();

}

}

**class** UserMainCode

{

**public** **static** String getMiddleCharacter(String str) //this

{

StringBuffer stringBuff=**new** StringBuffer();

**if**(str.length()%2==0) //4%2==0

{

stringBuff.append(str.substring(str.length()/2-1,(str.length()/2)+1));

//4/2 ---> 2-1=1 ,4/2=2+1 ---->3

//1,3-------> 1,2---------this t=0 h=1 i=2 s=3-------1,2 h,i

}

**return** stringBuff.toString();

}

}

**4.String processing – Long + Short + Long**

**Obtain two strings S1,S2 from user as input. Your program should form a**

**string of “long+short+long”, with the shorter string inside of the**

**longer String.**

**Include a class UserMainCode with a static method getCombo which accepts**

**two string variables. The return type is the string.**

**Create a Class Main which would be used to accept two Input strings and**

**call the static method present in UserMainCode.**

**Input and Output Format:**

**Input consists of two strings with maximum size of 100 characters.**

**Output consists of an string.**

**Refer sample output for formatting specifications.**

**Sample Input 1:**

**Hello**

**Hi**

**Sample Output 1:**

**HelloHiHello**

**Ans:**

**package** string;

**import** java.util.Scanner;

**public** **class** StringLongShortLong {

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

{

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

System.***out***.println("enter the word: ");//Hello

String letter=input.next();

System.***out***.println("enter the word: ");//Hi

String letter1=input.next();

System.***out***.println(*capitalStart*(letter,letter1));//---line no---16

input.close();

}

**public** **static** String capitalStart(String word1,String word2) //Hello Hi

{

StringBuffer sb=**new** StringBuffer();//

**int** word1Len=word1.length();//5

**int** word2Len=word2.length();//2

**if**(word1Len>word2Len)//5>2

{

sb.append(word1).append(word2).append(word1);//HelloHiHello

}

**else**// Hi>Hello----

{

sb.append(word2).append(word1).append(word2);//

}

**return** sb.toString();

}

}

enter the word:

Hai

enter the word:

Hello

HelloHaiHello

enter the word:

Hello

enter the word:

Hai

HelloHaiHello

**5.Strings Processing - Replication**

**Write a program to read a string and also a number N. Return the replica**

**of original string for n given time.**

**Include a class UserMainCode with a static method repeatString which**

**accepts the the string and the number n. The return type is the string**

**based on the problem statement.**

**Create a Class Main which would be used to accept the string and integer**

**and call the static method present in UserMainCode.**

**Input and Output Format:**

**Input consists of a string and integer.**

**Output consists of a string.**

**Refer sample output for formatting specifications.**

**Sample Input 1:**

**Lily**

**2**

**Sample Output 1:**

**LilyLily**

**Ans:**

**package** string;

**import** java.util.Scanner;

**public** **class** ReplicationString {

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

{

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

System.***out***.println("enter the word: ");

String word=input.next();

System.***out***.println("enter the count as you want: ");

**int** number=input.nextInt();

String result=*validString*(word,number);

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

input.close();

}

**public** **static** String validString(String word,**int** number)

{

StringBuffer sb=**new** StringBuffer();

**for**(**int** start=0;start<number;start++)

{

sb.append(word+" ");

}

**return** sb.toString();

}

}

enter the word: Kaviya

enter the count as you want: 3

Kaviya Kaviya Kaviya

enter the word: Lily

enter the count as you want: 4

Lily Lily Lily Lily

**6. Flush Characters**

**Write a program to read a string from the user and remove all the**

**alphabets and spaces from the String, and only store special characters**

**and digit in the output String. Print the output string.**

**Include a class UserMainCode with a static method getSpecialChar which**

**accepts a string. The return type (String) should return the character**

**removed string.**

**Create a Class Main which would be used to accept a string and call the**

**static method present in UserMainCode.**

**Input and Output Format:**

**Input consists of a strings.**

**Output consists of an String (character removed string).**

**Refer sample output for formatting specifications.**

**Sample Input :**

**cogniz$#45Ant**

**Sample Output :**

**$#45**

**Ans:**

**package** string;

**import** java.util.Scanner;

**public** **class** SpecialFlushCharacters {

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

{

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

System.***out***.println("enter the word: ");

String word=input.nextLine();

String result=*getValues*(word);

System.***out***.println("special characters: "+result);

input.close();

}

**public** **static** String getValues(String word)

{

StringBuffer sb=**new** StringBuffer();

**for**(**int** start=0;start<word.length();start++)

{

**char** letter=word.charAt(start);

**if**(!Character.*isAlphabetic*(letter))

sb.append(letter);

}

**return** sb.toString();

}

}

enter the word:

Kaviya@123#.com

special characters: @123#.

enter the word:

Cogni$#1@234zant

special characters: $#1@234

**7.Negative String**

**Given a string input, write a program to replace every appearance of the**

**word "is" by "is not".**

**If the word "is" is immediately preceeded or followed by a letter no**

**change should be made to the string .**

**Include a class UserMainCode with a static method “negativeString” that**

**accepts a String arguement and returns a String.**

**Create a class Main which would get a String as input and call the static**

**method negativeString present in the UserMainCode.**

**Input and Output Format:**

**Input consists of a String.**

**Output consists of a String.**

**Sample Input 1:**

**This is just a misconception**

**Sample Output 1:**

**This is not just a misconception**

**Sample Input 2:**

**Today is misty**

**Sample Output 2:**

**Today is not misty**

**Ans:**

package string;

import java.util.StringTokenizer;

import java.util.Scanner;

public class NegativeString {

public static void main(String[] args)

{

Scanner input =new Scanner(System.in);

System.out.println("enter the string : ");

String word=input.nextLine();

String result=validateNegative(word);

System.out.println(result);

input.close();

}

public static String validateNegative(String word) {

StringTokenizer st=new StringTokenizer(word," ");

StringBuffer sb=new StringBuffer();

while(st.hasMoreTokens())

{

String result=st.nextToken();

if(result.equals("is"))

{

sb.append(result.replace("is", " is not "));

}

else

sb.append(result);

sb.append(" ");

}

return sb.toString();

}

}

enter the string :

I think this is right time

I think this is not right time

enter the string :

Thank god! this is called luck!!

Thank god! this is not called luck!!

enter the string :

Today is misty

Today is not misty

**8. Name Shrinking**

**Write a program that accepts a string as input and converts the first two**

**names into dot-separated initials and printa the output.**

**Input string format is 'fn mn ln'. Output string format is 'ln [mn's**

**1st character].[fn's 1st character]'**

**Include a class UserMainCode with a static method getFormatedString which**

**accepts a string. The return type (String) should return the shrinked**

**name.**

**Create a Class Main which would be used to accept Input String and call**

**the static method present in UserMainCode.**

**Input and Output Format:**

**Input consists of a string.**

**Output consists of a String.**

**Refer sample output for formatting specifications.**

**Sample Input:**

**Sachin Ramesh Tendulkar**

**Sample Output:**

**Tendulkar R.S**

**Ans:**

package string;

import java.util.Scanner;

import java.util.StringTokenizer;

public class NameShrinkling {

public static void main(String[] args)

{

Scanner input=new Scanner(System.in);

System.out.println("enter the string: ");

String sentences=input.nextLine();

getValues(sentences);

input.close();

}

public static void getValues(String sentences) {

StringBuffer sb=new StringBuffer();

StringTokenizer st=new StringTokenizer(sentences," ");

String s2=st.nextToken();

String s3=st.nextToken();

String s4= st.nextToken();

sb.append(s4).append(" ");

sb.append(s3.substring(0,1));

sb.append(".");

sb.append(s2.substring(0,1));

System.out.println(sb);

}

}

enter the string:

Saraswthi Chinnakavan Kaviya

Kaviya C.S

enter the string:

Sachin Tendulker Ramesh

Ramesh T.S

**9.Start Case**

**Write a program to read a sentence in string variable and convert the**

**first letter of each word to capital case. Print the final string.**

**Note: - Only the first letter in each word should be in capital case in**

**final string.**

**Include a class UserMainCode with a static method printCapitalized which**

**accepts a string. The return type (String) should return the capitalized**

**string.**

**Create a Class Main which would be used to accept a string and call the**

**static method present in UserMainCode.**

**Input and Output Format:**

**Input consists of a strings.**

**Output consists of a String (capitalized string).**

**Refer sample output for formatting specifications.**

**Sample Input:**

**Now is the time to act!**

**Sample Output:**

**Now Is The Time To Act!**

**Ans:**

package string;

import java.util.Scanner;

import java.util.StringTokenizer;

public class StringStartCase

{

public static void main(String[] args)

{

Scanner input=new Scanner(System.in);

System.out.println("enter the sentences: ");

String sentence=input.nextLine();

System.out.println(capitalStart(sentence));

input.close();

}

public static String capitalStart(String s1)

{

StringBuffer sb=new StringBuffer();

StringTokenizer st=new StringTokenizer(s1," ");

while(st.hasMoreTokens())

{

String s2=st.nextToken();

String s3=s2.substring(0,1);

String s4=s2.substring(1,s2.length());

sb.append(s3.toUpperCase()).append(s4).append(" ");

}

return sb.toString();

}

}

enter the sentences:

Hope this is great year for me

Hope This Is Great Year For Me

enter the sentences:

i think this is the best version

I Think This Is The Best Version

**10.Occurance Count**

**Write a program to read a string that contains a sentence and read a**

**word. Check the number of occurances of that word in the sentence.**

**Include a class UserMainCode with a static method countWords which**

**accepts the two strings. The return type is the integer giving the count.**

**Note: The check is case-sensitive.**

**Create a Class Main which would be used to accept the two strings and**

**call the static method present in UserMainCode.**

**Input and Output Format:**

**Input consists of two strings.**

**Output consists of count indicating the number of occurances.**

**Refer sample output for formatting specifications.**

**Sample Input 1:**

**Hello world Java is best programming language in the world**

**world**

**Sample Output 1:**

**2**

**Sample Input 2:**

**hello world**

**World**

**Sample Output 2:**

**0**

**Ans:**

**package** string.occurencecount;

**import** java.util.Scanner;

**import** java.util.StringTokenizer;

**public** **class** Main {

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

{

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

String s1=input.nextLine();

String s3=input.next();

**int** count=0;

StringTokenizer st=**new** StringTokenizer(s1," ");

**while**(st.hasMoreElements())

{

String s2=st.nextToken();

**if**(s2.equals(s3))

count++;}

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

input.close();

}

}

package string.occurencecount;

import java.util.ArrayList;

import java.util.Collections;

import java.util.StringTokenizer;

public class UserMainCode

{

public static int empdis(String s,String f)

{

ArrayList<String> r=new ArrayList<String>();

int n;

StringTokenizer st=new StringTokenizer(s," ");

while(st.hasMoreTokens())

{

r.add(st.nextToken());

}

n=Collections.frequency(r, f);

return n;

}

}

**package** string.occurencecount;

**import** java.util.Scanner;

**public** **class** SubMain {

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

{

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

System.***out***.println("enter the string: ");//hello world this is kavi! how was the day??kavi hope ur doing well

String s=input.nextLine();

System.***out***.println("enter the word you want: ");//kavi

String f=input.next();

System.***out***.println(UserMainCode.*empdis*(s,f));

input.close();

}

}

enter the string:

hello world how are you?world

enter the word you want: world

2

**11.String Processing - III**

**Write a program to read a string where all the lowercase 'x' chars have**

**been moved to the end of the string.**

**Include a class UserMainCode with a static method moveX which accepts the**

**string. The return type is the modified string.**

**Create a Class Main which would be used to accept the string and call the**

**static method present in UserMainCode.**

**Input and Output Format:**

**Input consists of a string.**

**Output consists of a string.**

**Refer sample output for formatting specifications.**

**Sample Input 1:**

**xxhixx**

**Sample Output 1:**

**hixxxx**

**Sample Input 2:**

**XXxxtest**

**Sample Output 2:**

**XXtestxx**

**Ans:**

**package** string;

**import** java.util.Scanner;

**public** **class** Processing {

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

{

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

System.***out***.println("enter the word: ");//xxxxhiixx

String word=input.next();

String last=word.replaceAll("[x]", "");//hi

String last1=word.replaceAll("[^x]", "");//x x x x x x

System.***out***.println(last+last1);

input.close();}}

**Output:**

enter the word: xxHaixxKaviyaxx

HaiKaviyaxxxxxx

enter the word: XXtestxxhello

XXtesthelloxx

**Date Assigned :15:01:2023 Completed Date:18:01:2023**