Assignment No 9

1. Playing with String –

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

Program –

```
public class Program1
{
	public static void main(String[] args)
	{
		Program12 p12=new Program12();
		String out=p12.formString();
		System.out.println(out);
	}
}
```

```
import java.util.Scanner;
public class Program12
{
	String formString()
	{
		Scanner scanner=new Scanner(System.in);
		int len=scanner.nextInt();
		String[] array=new String[len];
		for(int j=0;j<len;j++)
		{
		array[j]=scanner.next();
		}
		int num=scanner.nextInt()-1;
		String ans="";
		for(int i=0;i<len;i++)
		{
		if (array[i].length()>num)
		{
		ans=ans+array[i].charAt(num);
		}
		else
		{
		ans=ans+"$";
		}
		}
		return ans;
	}
}
```

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

Program –

```
import java.util.Scanner;
public class ReverseString2
{
 public static String reverseSubstring(String str,int num1,int num2)
       {
        StringBuffer sb = new StringBuffer(str);
        sb.reverse();
        String s = sb.substring(num1, num1 + num2);
        return s;
       }
}


import java.util.Scanner;

public class ReverseString
       {
        public static void main(String[] args)
       {
        // TODO Auto-generated method stub
        Scanner sc=new Scanner(System.in);
        System.out.println("enter the values");
        String str=sc.nextLine();
```

```
                int num1=sc.nextInt();
                int num2=sc.nextInt();
                String str1=ReverseString2.reverseSubstring(str, num1,
                num2);
                System.out.println(str1);
                }
        }
```

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

Program -

```
import java.util.Scanner;
public class MidChar
{
  public static void main(String[] args)
  {
        // TODO Auto-generated method stub
        Scanner sc=new Scanner(System.in);
        System.out.println("enter the even string");
        String str=sc.nextLine();
        String str1=MidChar2.getMiddleChars(str);
        System.out.println(str1);
   }
}
```

```
public class MidChar2
{
public static String getMiddleChars(String str)
 {
   StringBuffer sb=new StringBuffer();
   if(str.length()%2==0)
   {
   System.out.println(str.substring((str.length()/2)-
 1,(str.length()/2)+1));
   }
   return sb.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

Program –

```
public class StringProcess2
{
public static String getCombo(String Long , String Short)
 {
 StringBuffer s5=new StringBuffer();
 int q=Long.length();
 int w=Short.length();
 if(q>w)
```

```
{
s5.append(Long).append(Short).append(Long);
}
else
{
s5.append(Short).append(Long).append(Short);
}
return s5.toString();
}
}
```

```
import java.util.Scanner;
public class StringProcess
{
     public static void main(String[] args) {
            // TODO Auto-generated method stub
            Scanner sc=new Scanner(System.in);
            System.out.println("enter the String");
            String s1=sc.nextLine();
            String s2=sc.nextLine();
            String s5=StringProcess2.getCombo(s1 , s2);
            System.out.println(s5);
            }
}
```

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

Program –

```
import java.util.Scanner;
public class StringReplication5
{
      public static String repeatString(String N)
      {
      Scanner sc=new Scanner(System.in);
      StringBuffer sb=new StringBuffer();
      String s=sc.next();
      int n=sc.nextInt();
      for(int i=0;i<n;i++)
      sb.append(s);
      return sb.toString();
      }
}
```

```
import java.util.Scanner;
public class StringReplicatin
{
     public static void main(String[] args)
     {
            // TODO Auto-generated method stub
            Scanner sc=new Scanner(System.in);
            String s=sc.next();
            String str=StringReplication5.repeatString(s);
            System.out.println(str);
      }
}
```

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

Program -

```
public class FlushChar2
{
  public static String getSpecialChar(String s1)
{
        int x=s1.length();
        StringBuffer sb=new StringBuffer();
        for(int i=0;i<x;i++){
        char c=s1.charAt(i);
        if(!Character.isAlphabetic(c))
        sb.append(c);
        }
        return sb.toString();
}
}
```

```
import java.util.Scanner;
public class FlushChar
{
    public static void main(String[] args)
    {
    Scanner in=new Scanner(System.in);
    String s1=in.nextLine();
    System.out.println(FlushChar2.getSpecialChar(s1));
    in.close();
    }
}
```

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

Program –

```
public class NegativeString2
{
  public static String negativeString(String s)
  {
  String newstring = "";
  int l = s.length();
  for(int i = 0; i < l; i++)
  {
  if(i-1 >= 0 && Character.isLetter(s.charAt(i-1))||
 i+2 < l && Character.isLetter(s.charAt(i+2)))
  {
  newstring += s.charAt(i);
  continue;
  }
  else if(i+1 < l && s.substring(i,
 i+2).equals("is"))
  {
  newstring += "is not";
  i++;
  } else
  newstring += s.charAt(i);
  }
  return newstring;
  }
}
```

```
import java.util.*;
public class NegativeString
{
 public static void main(String[] args)
 {
       // TODO Auto-generated method stub
        Scanner scanner=new Scanner(System.in);
        System.out.println("Enter the String:");
        String s=scanner.nextLine();
        String ans=NegativeString2.negativeString(s);
        System.out.println(ans);

 }
}
```

8. Write a program that accepts a string as input and converts the first two names into dot-separated initials and print a 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.

Program -

```
import java.util.StringTokenizer;
public class ShrinkName
{
   public static String getFormatedString(String s1) {
   StringBuffer sb = new StringBuffer();
   StringTokenizer st = new StringTokenizer(s1, " ");
   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));
   return sb.toString();
   }
}

import java.util.Scanner;
public class ShrinkName2
{
   public static void main(String[] args)
   {
         // TODO Auto-generated method stub
         Scanner sc = new Scanner(System.in);
         String s1 = sc.nextLine();
         String ans = ShrinkName.getFormatedString(s1);
         System.out.println(ans);
   }

}
```