**CIS 181 - Lab 1**

**Documenting Behavior**

**Overview**

For this lab you will be using the two sets of transformers: 1) sentence transformers ([SentenceTransformer.java](file:///C:\Users\Alan\Work\Teaching\CIS181\labs\lab1\SentenceTransformer.java)) and 2) word transformers ([WordTransformer.java](file:///C:\Users\Alan\Work\Teaching\CIS181\labs\lab1\WordTransformer.java)). Download them from lab1 section on myCourses.

Each transformer contains methods that allow it to perform certain tasks. In this lab you will assemble the components and invoke methods to convert given inputs to a specified output.

## Goals

1. To read Java methods to determine their function.
2. To document the behavior of methods from reading and execution behavior.
3. To solve problems by selecting and sequencing behaviors.

## Working with Transformers

The purpose of this set of exercises is to continue to think with objects. These exercises allow you to plan and implement solutions using an existing set of components. You are to determine how to organize and invoke components and behaviors to accomplish a task.

Consider the **example** below. Your are given two strings as input:

* String myString1="Al says hello there.";
* String myString2="Who is this?";

You are asked to produce the string ***Al says who is there?***. This is done by choosing components from the classes you have been given, then sending these components messages that process the input strings, creating the new strings you need to assemble for the requested result.

String myString1="Al says hello there.";

String myString2="Who is this?";

//there

String temp1=myString1;

temp1=SentenceTransformer.depunctuate(myString1);

for (int i=1;i<4;i++) temp1=SentenceTransformer.lastWords(temp1);//temp1="there"

temp1=WordTransformer.leftPadder(temp1);// there

//Al says

String temp2;

temp2=SentenceTransformer.firstWord(myString1);//Al

temp3=SentenceTransformer.lastWords(myString1);

temp3=SentenceTransformer.firstWord(temp3);

temp2=WordTransformer.rightPadder(temp2,WordTransformer.leftPadder(temp3));

// who is there?

String temp4;

temp4=SentenceTransformer.firstWord(myString2);//Who

temp4=WordTransformer.lowerCase(temp4);//who

temp4=WordTransformer.leftPadder(temp4);// who

String temp5=SentenceTransformer.lastWords(myString2);//is this?

temp5=SentenceTransformer.firstWord(temp5);//is

temp5=WordTransformer.leftPadder(temp5);// is

temp4=WordTransformer.rightPadder(temp4,temp5);

temp4=WordTransformer.rightPadder(temp4, temp1);

temp4=WordTransformer.rightPadder(temp4, "?");

//Al says who is there?

temp2=WordTransformer.rightPadder(temp2,temp4);

System.out.println(temp2);

The final punctuation on the phrases is produced as shown in the example. Since there is no method given to extract a character as a string.

Before you can begin the exercises, you must have an understanding of the behavior of the methods in the two classes. You will notice all the methods are ***static***, so they can be invoked using classname.methodname() as I did in the example. Read the methods in the classes then prepare and execute experiments to confirm you understanding of their behavior, then write documentation for each method describing the nature of the parameters and the result the method produces. Your work should be recorded in the Documentation deliverable for Lab 1.

## Exercises

After you are familiar with the above example. The exercises for this lab are in the portfolio under **Lab1 Exercises (word file)**. For each task the input and expected output are provided. You should plan a solution, implement it using Eclipse (or other Java IDE), check the solutions against the expected output, then put (paste) your solution in the space provided in the form. At the end of lab you should submit your solutions. You may **discuss with your classmates (using online meeting or keeping social distancing)** on these exercises, but each person must submit **your own code developed by yourself**.

You should think carefully when introducing new variables and extracting information from values. You need to plan your strategy carefully, trying to make effective use of the resources. Too many variables become hard to keep track of. Furthermore, you may wish to use documentation to help you track what results you have. You are allowed to test your progress by outputting values using System.out.println() at your discretion.