# Lab Assignment 8\_2 – Text Processing

## Part 1:

1. Create a new Java project called **Lab8\_2A** and a class named **Lab8\_2A**
2. Create a secondary class called **SentenceParser**.
3. Have the following instance variables in this class.
   1. A string variable named sentence
   2. An array of strings named wordArray. (Only put the first half of the array declaration here.)
4. The constructor (no parameters) should do the second half of the **wordArray** declaration. **wordArray** will have 10 elements.
5. Create a void method named **readFile** with no parameters. It should set up the text file and read data from it.  
   For each line in the text file:
   1. Read the string (which will have spaces) into **sentence**
   2. Print **sentence**
   3. Call the **parseIt** method
   4. Call the **printEvens** method
   5. Call the **blankArray** method
6. Create a void method named **parseIt** (no parameters) that will
   1. Declare and create a String Tokenizer object, sending **sentence** in as the parameter (The delimiters will be spaces.)
   2. Use a while has more tokens loop to place each token into **wordArray**.
7. Create a void method named **printEvens** (no parameters) that will loop through the array and print the items with even array indexes (0, 2, 4, …)  
   They should all be printed on one line; then print a blank line.
8. Create a void method named **blankArray** (no parameters) that will set every spot in the array to an empty string (“”).
9. Back in the main class:
   1. Declare and instantiate a new **SentenceParser** object named **robbie**.
   2. Call the **readFile** method for **Robbie** (no parameters).

## Part 2:

1. Create a new Java project called **Lab8\_2B** and a class named **Lab8\_2B**
2. Create a secondary class named **TextEditor** with the following:
   1. A String instance variable named **message**
   2. A constructor with one String parameter that sets **message** equal to the parameter
   3. A void method named **replaceWord** (no parameters) that will ask the user for an old word and a new word. The method should replace all the instances of old word with new word in **message**.  
      (Look in the PowerPoint notes for the replaceAll String method.)
   4. A void method named **editAndPrint** (no parameters) that will print **message**. Use a loop to look at each character in the string and print it with the following changes.
      1. The first character is a letter and should be changed to upper case.
      2. If the current character is a period (.), print that character, then a space, and then print the next character in upper case. (The next character will either be a letter or a space; printing it in “upper case” won’t hurt anything if it’s a space.)
      3. If the current character is a %, then don’t print it, but go to the next line twice, so that you leave a blank line. The print the next character as an upper case character.
      4. If none of the above cases are true, then just print the current character.
3. Back in the main method of class Lab8\_2B do the following.
   1. Read the whole line of text from the text file (**Lab8\_2B.txt**) and put it into a String variable
   2. Create a **TextEditor** object sending the String variable as a parameter for it.
   3. Call **replaceWord** for your object.
   4. Call **editAndPrint** for your object.

(When you run the program, if you enter **star** for the old word and **sun** for the new word, your output should look like the following. I will grade it with a different text file though.)

Twinkle, twinkle, little sun. How I wonder what you are.

Up above the world so high, like a diamond in the sky.

When the blazing sun is gone, when he nothing shines upon.

Then you show your little light. Twinkle, twinkle, all the night.

Twinkle, twinkle, little sun. How I wonder what you are.