**Computer Science 3033  
Due: March 10, 2004**

**Lab - The Flesch Readability Index**

**Introduction:**

The Flesch readability index is a tool for estimating the reading comprehension level necessary to understand a written document. For a given document, the Flesch readability index is an integer indicating how difficult the document is to understand, with lower numbers indicating greater difficulty. For example, the table below shows typical Flesch readability index values for some common(and some not-so-common) reading material:

| **Material** | **Flesch Index** |
| --- | --- |
| Comics | 95 |
| Consumer Ads | 82 |
| Sports Illustrated | 65 |
| Time | 57 |
| New York Times | 39 |
| Auto Insurance | 10 |
| IRS Code | -6 |

Flesch readability indexes are also often translated into the educational level that is usually necessary to understand a document:

| **Flesch Index** | **Educational Level** |
| --- | --- |
| 91-100 | 5th grade |
| 81-90 | 6th grade |
| 71-80 | 7th grade |
| 66-70 | 8th grade |
| 61-66 | 9th grade |
| 51-60 | High School |
| 31-50 | Some College |
| 0-30 | College Graduate |
| < 0 | Law School Graduate |

The Flesch readability index for a document is computed using 5 steps:

* 1. Count the number of words in the document.
  2. Count the number of syllables in the document.
  3. Count the number of sentences in the document.
  4. Compute the index as:

Syllables Words

Flesch Index = 206.835 - 84.6 \* --------- - 1.015 \* ---------

Words Sentences

* 1. Round the index to the nearest integer.

With a simple method for computing the Flesch readability index it is possible for authors to write and revise documents until they are comprehensible to the target audience. For example, a newspaper reported might continue to revise an article until the Flesch readability index is above 60.

At least one popular word processor is able to compute the Flesch readability index. If you [ask Microsoft Word to display the readability statistics for a document](http://enrollmentoptions.sandi.net/workshops/walklikealibrarian/evaluatesites/wordhelp.html), it will display a Flesh reading ease score, which is the same as the readability index. If you are interested in more information see [chapter 2, of Flesch's "How to Write Plain English"](http://www.mang.canterbury.ac.nz/courseinfo/AcademicWriting/Flesch.htm) (interestingly enough, Flesch was French!)

**Setup:**

You will also need to download the [TextFileReader.class](http://www.mtsu.edu/~csjudy/3033/Labs/TextFileReader.class) file to your disk. You will use a TextFileReader object to read a document from a text file in order to compute its Flesch readability index. There are more details on how to use the TextFileReader class below.

**Assignment:**

In this lab you will develop and test two classes and write an application that uses your classes to compute the Flesch readability index for a document. The first class you will write and test is the Word class which represents a single word and provides a method for counting the number of syllables in a word. The second class that you will write and test is the Sentence class. The Sentence class represents a collection of words. It provides methods for counting the number of words in the sentence and extracting each word from the sentence one by one. With both of these classes written and tested you will write an application (i.e. a class with a main method) which uses Word and Sentence objects to compute the Flesch readability index of a document. More details on each part of this assignment appear in the following sections:

**The Word Class**

Implement the Word class according to the description which appears in the [Java documentation for the Word class](http://www.mtsu.edu/~csjudy/3033/Labs/Word.html). Add any additional methods that you deem necessary or that should be added to a Java class. Notice that the rules for how syllables should be counted are described in the documentation for the countSyllables method.

Write a test program for your Word class in the file WordTest.java. This test program should contain only a main method. The main method should create a number of Word objects and test each of the instance methods. Be sure to select test cases which are sufficient to fully test your Word class.

**The Sentence Class**

Implement the Sentence class according to the description which appears in the [Java documentation for the Sentence class](http://www.mtsu.edu/~csjudy/3033/Labs/Sentence.html). Add any additional methods that you deem necessary or that should be added to a Java class. The easiest approach to implementing the Sentence class is to use a [StringTokenizer](http://java.sun.com/j2se/1.3/docs/api/java/util/StringTokenizer.html) as part of the instance data. Notice that the Sentence class is coupled to the Word class. Thus, your Word class must be completed before you can write the Sentence class. Write a test program for your Sentence class in the file SentenceTest.java.

**The Flesch Appplication**

In a class named Flesch, write a main method that uses your Sentence and Word classes to compute and display the Flesch readability index of a document. Your Flesch application should accept the name of the file to be analyzed on the command line and then compute and display the Flesch readability index for that file.

In order to read the input file you will need to use a TextFileReader object. The [Java documentation for the TextFileReader class](http://www.mtsu.edu/~csjudy/3033/Labs/TextFileReader.html) contains an example which should be sufficient to get you started.

You should create a few simple text files to test your Flesch program. If you make these files simple enough you can compute the index by hand. That way, when you run you program you can tell if it computes the correct answer. After you do that you might want to run your program on some more extensive text files. The [Etexts web site](http://www.infomotions.com/etexts/) provides an extensive library of text files containing public domain literary works. Some examples, with their readability index include:

| **Work** | **Flesch Index** |
| --- | --- |
| [Romeo & Juliet](http://www.infomotions.com/etexts/literature/english/1500-1599/shakespeare-romeo-48.txt) | 84 |
| [Tom Sawyer, Detective](http://www.infomotions.com/etexts/literature/american/1900-/twain-tom-40.txt) | 84 |
| [Beowulf](http://www.infomotions.com/etexts/literature/english/700-799/anonymous-beowulf-543.txt) | 75 |
| [Plato's Republic](http://www.infomotions.com/etexts/philosophy/400BC-301BC/plato-republic-762.txt) | 69 |

To test your program on one of these texts, click on its link and then choose "Save as..." from the "File" menu. Be sure to delete these files when you are finished to avoid filling up your disk.

**Grading**

This lab will be worth 100 points. The Word and Sentence classes will be worth 30 points each. The WordTest and SentenceTest classes will be worth 10 points each. The Flesch program will be worth 20 points.

On the due date for this lab, you must hand in a printout of the following files:

* + Word.java
  + WordTest.java
  + Sentence.java
  + SentenceTest.java
  + Flesch.java

These files must also appear on your disk. I will be testing your program on files of my choice.