EECS 293

Software Craftsmanship

2013 Fall Semester

Programming Assignment 1

Due at your recitation session on September 2-6, 2013

Reading

Read Chapters 1, 2, 3, and 4 in the textbook.

Programming

Objective: Review of programming and data structures

The objective of this assignment is to segment a text into tokens. You can assume that the text contains only printable characters. Each token is a string of length *k* or less. Additionally, blanks (along with newlines, carriage returns) and punctuation characters (“,.!?”) must be included in a token of unit length. For example, the text “I came, I saw, I left.” can be segmented by *k*=3 as `I` ` ` `cam` `e` `,` ` ` `I` ` ` `saw` `,` ` ` `I` ` ` `lef` `t` `.` . Additionally, the segmentation can be expressed numerically, where each index represents the place at which each segment occurs *for the first time* in the text. For example, “I came, I saw, I left.” becomes 1 2 3 4 5 2 1 2 6 5 2 1 2 7 8 9 with the legend: `I`:1, ` `: 2, `cam`: 3, `e`: 4, `,`: 5, `saw`: 6, `lef`: 7, `t`: 8, `.`: 9.

Implement a `class Segmentator` with the following methods:

- A constructor `Segmentator(Integer segment_length, String filename)` that creates a Segmentator associated to the given segment length *k* and filename. It throws no exception.
- A `List<String> segment()` method that segments the given file. It throws an appropriate exception in case of I/O or other error.
- A `List<Integer> segment()` method that segments the given file numerically. It throws an appropriate exception in case of I/O or other error

Additionally, implement a `main` that, given a segment length and file name in the command line, prints the segments (Strings) separated by newlines.

The `Segmentator` class must be documented in JavaDoc. Test cases must be created in JUnit to test all its public methods. Most students have seen JavaDoc and JUnit in EECS

132 or 233: if you did not, it is your responsibility to catch up. The code must compile and run.

The class Segmentator will be adapted or reused in future programming assignments.

If you are in a non-Java section, it is your responsibility to adapt these specifications to your programming language.

Submission

Bring a copy to recitation to display on a projector. Additionally, submit an electronic copy of your program to blackboard.

Grading

This programming assignment is required but does not contribute to your grade.