EECS 293
Software Craftsmanship
2013 Fall Semester

Programming Assignment 7

Due at your recitation session on October 14-18

Programming

In this programming assignment, you will implement the compression method.

First, revise your pseudo-code to reflect the discussion during recitation.

Second, follow the instructions to clone the Git repository that has been setup eecslinab6. The instructions are posted as part of the Git lecture notes on Blackboard.

Third, create a new repository ‘compression.git’. Place all of your homework artifacts in the compression.git repository.

Fourth, implement the compression method according to the pseudo-code programming process. You should reuse the segmentator that you wrote for the first assignment. Follow your revised pseudo-code faithfully, even if you can think of additional improvements.You will probably need to implement some methods for error-handling. However, since compression is primarily for hypothetical future use as a component in a larger project, your code may contain only simple stubs for error-handling.

Fifth, make sure during development you make small regular commits. When you are finished with your homework remember to tag your release and push that tag to your git repository on eecslinab6.

```
$ git tag -a hw7
$ git push --tags
```

Discussion Guidelines

The first part of the class discussion is on git: the recitation leader will pull your changes from your repository on eecslinab6. You will then give the class a brief overview of your development process using the commit history stored in git. The aim is to have relatively small self-contained commits with descriptive commit messages.

The bulk of the discussion will focus on the pseudo-code programming process (Chapter 9): appropriateness and completeness of pseudo-code for rapid implementation, pseudo-code comments in final code, etc. For the sole purpose of comparing with the previous assignment, leave all pseudo-code comments in your code.

Evaluation Guidelines

At this stage of the course, a more strict view will be taken of the following basic issues:

- Routines with McCabe’s complexity exceeding 4
- Improperly named routines
- Repeated code